Argumentation-based Reasoning about Plans, Maintenance Goals and Norms

ZOHREH SHAMS*, University of Bath, UK
MARINA DE VOS, University of Bath, UK
NIR OREN, University of Aberdeen, UK
JULIAN PADGET, University of Bath, UK

In a normative environment an agent's actions are not only directed by its goals, but also by the norms activated by its actions and those of other actors. The potential for conflict between agent goals and norms makes decision-making challenging, in that it requires looking-ahead to consider the longer term consequences of which goal to satisfy or which norm to comply with in face of conflict. We therefore seek to determine the actions an agent should select at each point in time taking account of its temporal goals, norms and their conflicts. We propose a solution in which a normative planning problem is the basis for practical reasoning based on argumentation. Various types of conflict within goals, within norms and between goals and norms are identified based on temporal properties of these entities. The properties of the best plan(s) with respect to goal achievement and norm compliance are mapped to arguments, followed by mapping their conflicts to attack between arguments, all of which are used to identify why a plan is justified.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Distributed artificial intelligence**; **Intelligent agents**;

Additional Key Words and Phrases: Argumentation, Goals, Norms

ACM Reference Format:
Zohreh Shams, Marina De Vos, Nir Oren, and Julian Padget. 2018. Argumentation-based Reasoning about Plans, Maintenance Goals and Norms. *ACM Trans. Autonom. Adapt. Syst.* 1, 1, Article 1 (January 2018), 39 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In many environments, determining which actions to take – or *practical reasoning* – requires autonomous entities (agents) not only to consider the goals they wish to achieve, but also to take norms into account. Norm is interpreted in several ways in the literature: some interpretations capture general intentions about what is desirable or not (e.g.; thou shalt not steal[1]), others indicate when a real-world action has institutional effect [Giannikis and Daskalopulu 2011; Li 2014] (e.g.; those passing the red light will be fined), while others capture precise descriptions of contextualized behaviour [Kollingbaum 2005; Pacheco 2012] (e.g.; it is obligatory to pay the fine within two weeks of receiving it). The key differentiator is that some norms offer general advice about what is (not) good, but not how to achieve it, while others provide specific advice about how to act in the context of a more general (sometimes implicit) norm. The latter category are referred to as *the regulatory norm*. In the work presented here, we use the pragmatic notion of norm,

*Present Affiliation: Department of Computer Science and Technology, University of Cambridge, UK (CB3 0FD)
[1]This "norm" dates back to Saint Thomas Aquinas, 13th century.

Authors' addresses: Zohreh Shams, University of Bath, Bath, UK, zohreh.shams@cl.cam.ac.uk; Marina De Vos, University of Bath, Bath, UK, m.d.vos@bath.ac.uk; Nir Oren, University of Aberdeen, Aberdeen, UK, n.oren@abdn.ac.uk; Julian Padget, University of Bath, Bath, UK, j.a.padget@bath.ac.uk.

the regulatory norm, by advising an agent what it is obliged to do (i.e., *obligation norm*), or forbidden from doing (i.e., *prohibition norm*), either of which it may violate. Our focus is on *action-based* regulatory norms [Meneguzzi et al. 2015; Oren et al. 2011], while state-based regulatory norms (i.e.; norms that oblige or forbid certain state of affairs) have also been discussed in the literature [De Vos et al. 2013; Kollingbaum 2005].

Norm enforcement approaches (e.g., [Pitt et al. 2013; y López et al. 2005]) often associate punishment with norm violation, so that avoiding punishment incentivises agents to comply with norms. However, agents might not wish or be able to comply with all norms imposed on them, due either to norm conflicts[2], or to their wishing to achieve a goal that outweighs the cost of non-compliance. In such cases, the agent must identify which norm(s) it might violate, and accept the punishment or lack of reward for so doing, or which goals it might not satisfy. In identifying actions, long-term impacts must be considered. For instance, a goal satisfied at the cost of violating a norm might hinder or prevent more important goals from being achievable. One way of reasoning about such long-term impacts of compliance and violation requires the agent to consider the conflicts in the context of plans available to it. In so doing, an agent can take into account the benefit of goal achievement and norm compliance against the cost of goal failure and norm violation in different plans, and hence determine which plans are best to follow in presence of conflict.

Norms have been considered in both planning and plan selection in the past [Belchior et al. 2018; Broersen et al. 2002; Kollingbaum 2005]. In order to generate conflict-free plans these works aim at resolving all normative conflicts. Since norms cannot be violated, if conflict resolution is not possible, the planning fails. While generating conflict-free plans, due to the possibility of violation, we allow more freedom in action selection, where an agent can take an action that may cause a normative conflict. Alternative ways of resolving conflict then give rise to different plans. Our departure point in doing this is [Shams et al. 2016], which analyses norm-goal and goal-goal conflicts statically, considering whether they require different states of affairs to hold, but not the times at which the norms and goals are active. Here, we extend that model by (i) considering temporal properties for goals, and (ii) proposing temporal solutions to goal-goal and goal-norm conflicts as well as norm-norm conflicts.

Apart from the above distinction, earlier work mentioned [Belchior et al. 2018; Broersen et al. 2002; Kollingbaum 2005; Panagiotidi et al. 2012] focus on the reasoning processes of a fully autonomous system, without considering the explainability of the system. In contrast, we consider domains where humans may need to understand why some action or plan was selected for execution (such as human-agent teams, or where a developer is debugging agent behaviour). This requires a transparent reasoning mechanism, rather than the numerical utilities of [Broersen et al. 2001; Kollingbaum and Norman 2003], that can serve as the basis for the justification of agent behaviour. We utilise formal argumentation to derive such a reasoning process.

Argumentation serves as an effective computational tool for various agent activities including agent reasoning [Amgoud 2003; Bench-Capon et al. 2009; Dung 1995; Gaertner and Toni 2007; Oren et al. 2007]. It supports the derivation of consistent conclusions from conflicting, inconsistent and incomplete information as generic arguments (or *argument schemes* [Walton 1996]). Such schemes also allow us to capture conflicts between arguments through the use of *critical questions*, representing the context in which an argument is invalid. We can then use argumentation semantics [Dung 1995] to determine plan *justifiability* with respect to goal satisfaction and norm compliance/violation. In addition, we describe how to discriminate between justified plans and identify a most preferred set of plans. Finally, we investigate the formal properties of what our approach considers the best plan(s).

[2]This may well be due to the fact that norms come from different authorities aiming at regulating different aspects of agent behaviour.

The remainder of this paper is structured as follows. In the next section, we provide an illustrative scenario to motivate our approach throughout the paper. In Section 3 we present the syntax and semantics of a model for normative practical reasoning. Section 4 then explains the procedures to identify the best plan(s) using argumentation techniques. The formulation of the illutrative scenario in Section 2, is provided in Section 5, along with examples for identifying the best plan in this scenario. Related works are surveyed in Section 6, followed by conclusion in Section 7.

2 ILLUSTRATIVE SCENARIO: PART I

To illustrate our approach we extend the scenario in [Shams et al. 2017], such that like norms, goals have temporal properties. Here a software agent acts as a supervisory system in a disaster recovery mission and supports human decision-making. The agent plans on behalf of a group of human actors who are in charge of responding to an emergency caused by an earthquake. It monitors the situation (e.g.; contamination of water, detection of aftershocks, etc.) to advise humans of the different courses of action available, and help them decide which is best. We set the following goals:

(1) Running a hospital to help wounded people: fulfilled when, between timesteps[3] 5 and 8, medics are present to offer help and they have access to water and medicines.
(2) Organising a survivors' camp: fulfilled when the camp area is secured and a shelter is built by timestep 15.

We impose the following three norms that the agent must consider while devising plans to satisfy the above goals:

(1) It is forbidden to build a shelter within 3 time units of detecting shocks.
(2) It is obligatory to stop water distribution for 2 time units once contamination is detected in the water.
(3) It is forbidden to stop water distribution within 3 time units of building a shelter.

The full formulation of this scenario is provided in Section 5.

3 A MODEL FOR NORMATIVE PRACTICAL REASONING

We start from the model in [Shams et al. 2016] and [Shams et al. 2017], to build a richer model for practical reasoning. In [Shams et al. 2016] a STRIPS [Fikes and Nilsson 1971] like action and planning language is the basis for normative practical reasoning, allowing for (i) durative actions to reflect the time it takes to execute actions; (ii) a set of multiple potentially inconsistent goals; and (iii) normative considerations in the practical reasoning process. The contribution of this article is to extend the model in [Shams et al. 2017] to deal with goals that have temporal properties. As a consequence, the contribution is also in formulating the conflict within goals, within norms and between the two with respect to the newly defined temporal properties for goals. Such temporal goals [Hindriks et al. 2009] are discussed in detail in agent programming languages such as GOAL [de Boer et al. 2007], however they are not commonly used in practical reasoning frameworks due to the challenges they introduce to the agent's decision making. The syntax and semantics of the model are presented in the sections that follow.

3.1 Syntax and Semantics

A normative practical reasoning problem describes the domain over which reasoning takes place.

Definition 3.1 (Normative Practical Reasoning Problem). A normative practical reasoning problem is a tuple $P = (FL, \Delta, A, G, N)$ where

[3]Time units may correspond to whatever is appropriate for the scenario, e.g., hours or days.

- FL is a set of fluents;
- Δ is the initial state;
- A is a finite, non-empty set of durative actions;
- G is the agent's set of goals;
- N is a set of norms imposed on the agent establishing what the agent is obliged or forbidden to do under certain conditions.

3.1.1 Fluents and Initial State. FL is a set of domain fluents that accounts for the description of the domain the agent operates in. A literal l is a fluent or its classical negation i.e., $l = fl$ or $l = \neg fl$ for some $fl \in FL$, with $\neg\neg f = f$ for $f \in FL$. For a set of literals L, we define $L^+ = \{fl \in FL | fl \in L\}$ and $L^- = \{fl \in FL | \neg fl \in L\}$ to denote the fluents in L that are positive or negative, respectively. L is *well-defined* if $L^+ \cap L^- = \emptyset$.

The semantics of the normative practical reasoning problem are defined over a set of states Σ. A state $S \subseteq FL$ is determined by the set of fluents that hold *true* at a given time, while the other fluents (those that are not present) are considered to be *false*. A state $S \in \Sigma$ *satisfies* fluent $fl \in FL$, denoted as $S \models fl$, if $fl \in S$. It satisfies its negation, denoted as $S \models \neg fl$, if $fl \notin S$. This notation can be extended to a set of literals as follows: set X is satisfied in state S, denoted $S \models X$, when $\forall x \in X \cdot S \models x$.

The set of fluents that hold in the initial state is denoted as $\Delta \subseteq FL$.

We use an explicit representation of time, where time $t + 1$ is one unit ahead of t. Thus, a state S_t at a certain time t refers to the set of fluents that hold at that time t.

3.1.2 Durative Actions. A is a set of durative actions, each of which has pre- and post-conditions. The effects of an action (captured by its post-conditions) are not immediate, that is, it takes a non-zero period of time for the effects of an action to take place, during which the action is said to be *in progress*. We also assume that actions are deterministic (i.e., execution of an action in a given state leads to one particular state) and take a fixed amount of time. We also assume the preconditions of an action hold while the action is in progress.

Definition 3.2 (Durative Actions). A durative action $a \in A$ is a tuple (pr, ps, d) where $pr, ps \subseteq FL \cup \neg FL$ such that $pr^+ \cap pr^- = ps^+ \cap ps^- = \emptyset$ and $d \in \mathbb{N}^+$.

The sets pr,ps are the possibly empty well-defined pre- and post-conditions of the action while d is the duration of the action. For an action $a = (pr, ps, d)$ we define the projections $pr(a), ps(a)$ and $d(a)$ to access the values pr, ps, d respectively. Moreover, we use $pr(a)^+$ and $pr(a)^-$ to refer to fluents that retrospectively should or should not hold in the precondition $pr(a)$; similarly, $ps(a)^+$ and $ps(a)^-$ refer to the positive and negated fluents in $ps(a)$. Also as mentioned earlier we use an explicit and discrete representation of time. Thus if an action has a duration d and its starting point is t, then its ending point will be $t + d$.

An action a can be executed in a state S if its preconditions hold in that state (i.e., $S \models pr(a)$). The postconditions of a durative action cause changes to the state T in which the action ends. These changes consist of adding the positive postconditions $ps(a)^+$ to T and deleting the negative postconditions from $ps(a)^-$ from T. As a result, for a state T in which action a ends, we have: $T \models ps(a)^+$ and $T \not\models ps(a)^-$.

3.1.3 Goals. Goals identify the state of affairs in the world that an agent wants to satisfy. Different types of goals and their characteristics have been classified in the literature [van Riemsdijk et al. 2008]. Here, we consider maintenance goals where a state has to be brought about first and then maintained over a certain period[4].

Definition 3.3 (Goals). A goal $g \in G$ is a tuple $(requirement, maintenanceStart, maintenanceEnd)$, where $requirement \subseteq FL \cup \neg FL$ such that $requirement^+ \cap requirement^- = \emptyset$ and $maintenanceStart, maintenanceEnd \in \mathbb{N}^+$ such that $maintenanceStart < maintenanceEnd$.

The well-defined set of literals *requirement* represents the positive and negative goal requirements of the goal while *maintenanceStart* and *maintenanceEnd* encode the start and end of the maintenance period. For a goal $g = (requirement, maintenanceStart, maintenanceEnd)$ we define the projections $r(g), m_s(g), m_e(g)$ to obtain the values *requirement*, *maintenanceStart*, *maintenanceEnd*, respectively, from the tuple g. Intuitively, a goal is satisfied if $S_i \models r(g)$ for all states S_i with i between m_s and m_e.

3.1.4 Norms. As noted in the introduction, we model the notion of action-based regulatory norm to capture what an agent *ought* (not) to do and what an agent is *not permitted* to do. The role of an obligation is to motivate the agent to execute a specific action and the role of prohibition is to inhibit the agent from executing a particular action. Is it worth noting that we are considering norms from the point of view of an agent - any norm the agent is aware of is assumed to apply to it.

Definition 3.4 (Norms). A norm is a tuple of the form $n = (op, a_c, a_s, dl)$, where

- $op \in \{o, f\}$ is the deontic operator determining the type of norm, which is an obligation or a prohibition;
- $a_c \in A$ is the durative action (cf. Def. 3.2) that activates the norm;
- $a_s \in A$ is the durative action (cf. Def. 3.2) that is the subject of the obligation or prohibition;
- $dl \in \mathbb{N}$ is the norm deadline relative to the activation condition, which is the completion of the action a_c.

An obligation expresses that executing action a_c obliges the agent to *start* the execution of a_s within dl time steps of the end of execution of a_c. If this occurs, the obligation is complied with, otherwise, it is violated. A prohibition expresses that executing action a_c prohibits the agent to start the execution of a_s within dl time units of the end of execution of a_c. Such a prohibition is complied with if the agent *does not start* executing a_s before the deadline and is violated otherwise. Note that since actions cannot be interrupted whilst in progress, the start of execution of a_s during compliance period counts as compliance for obligations, the reverse applies to prohibitions.

A norm can be activated multiple times in a sequence of action, generating different instances of the original norm [Alan S. Abrahams and Jean M. Bacon 2002]. Below we define instantiated norms in which instead of deadline, we have the start and end of compliance period.

Definition 3.5 (Instantiated Norm). Given a norm $n = (op, a_c, a_s, dl)$ and the occurrence of action a_c at time t_{a_c}, we define n's instantiation as $n = (op, a_c, a_s, comp_s, comp_e)$ where $comp_s$ and $comp_e$ mark the start and end of compliance period, respectively. The compliance period is calculated based on the start time t_{a_c} of action a_c and its duration (i.e.; $d(a_c)$), as follows: $comp_s = t_{a_c} + d(a_c)$, and $comp_e = t_{a_c} + d(a_c) + dl$.

Having explained the syntax of the model, in the next section we turn our attention to the semantics.

[4]Note that requiring some condition always to be maintained, e.g., keeping the temperature above freezing, could be viewed as a maintenance goal from time zero to infinity.

3.2 Plans and their Properties

Here we begin by defining a sequence of actions and what makes it a plan.

Definition 3.6 (Sequence of Actions). Suppose $P = (FL, \Delta, A, G, N)$ is a normative practical reasoning problem. $\pi = \langle (a_0, 0), \cdots, (a_n, t_{a_n}) \rangle$ is a sequence of actions, where $a_i \in A$, $n \geq 0$, and $t_{a_i} \in \mathbb{N}^+$. The action-time pair (a_i, t_{a_i}) reads as action a_i is executed at time t_{a_i}. We assume that π is temporally ordered, that is, $\forall i < j \cdot t_{a_i} < t_{a_j}$.

The total duration of a sequence of actions, defined as $Makespan(\pi)$, is given by Equation 1.

$$Makespan(\pi) = max(t_{a_i} + d(a_i) : 0 \leq i \leq n) \tag{1}$$

For a sequence of actions π, we denote the set of action-time pairs for which the actions end at time k as $A(k)$, where $A(k) = \{(a, t) \in \pi \mid k = t + d(a)\}$.

Sequences of actions start at specific state, which will then change depending on the actions executed: the postconditions of the executed actions cause changes to the state in which the actions end.

Definition 3.7 (Sequence of States). Let $\pi = \langle (a_0, 0), \cdots, (a_n, t_{a_n}) \rangle$ be a sequence of actions for the normative practical reasoning problem $P = (FL, \Delta, A, G, N)$. Execution of π from a state S_0 results in a sequence of states $S(\pi) = \langle S_0, \cdots S_m \rangle$, with $m = Makespan(\pi)$, such that:

$$\forall k > 0 \cdot \; S_k = \begin{cases} (S_{k-1} \setminus \bigcup\limits_{(a,t) \in A(k)} ps(a)^-) \; \bigcup \; (\bigcup\limits_{(a,t) \in A(k)} ps(a_i)^+) & A(k) \neq \emptyset \\ S_{k-1} & A(k) = \emptyset \end{cases} \tag{2}$$

Conflict caused by time, known as a *concurrency conflict*, prevents some actions from being executed in an overlapping period of time. We define conflicting actions following [Blum and Furst 1997], as two actions that have pre- and postconditions that contradict each other. This heuristic eliminates the possibility of two actions that are likely to conflict due to their contradictory pre- and postconditions, to be executed concurrently. [5]

Definition 3.8 (Conflicting Actions). The set of action pairs that have a concurrency conflict, denoted as cf_{action}, is defined as:

$$\begin{aligned} cf_{action} = \{(a_i, a_j) \; s.t. \; a_i, a_j \in A \mid & \exists \; r \in pr(a_i) \cup ps(a_i)^+ \cdot \neg r \in pr(a_j) \cup ps(a_j)^- \text{ or} \\ & \exists \; \neg r \in pr(a_i) \cup ps(a_i)^- \cdot r \in pr(a_j) \cup ps(a_j)^+ \} \end{aligned} \tag{3}$$

Example 3.9. Take action $secure = \left(\{evacuated\}, \left\{ \begin{array}{c} areaSecured, \\ noAccess \end{array} \right\}, 5 \right)$. The pre and postconditions of this action are inconsistent with the pre and postconditions of action $buildShelter = \left(\left\{ \begin{array}{c} areaSecured, \\ evacuated \end{array} \right\}, \left\{ \begin{array}{c} shelterBuilt, \\ \neg evacuated \end{array} \right\}, 2 \right)$. Therefore, these two actions cannot be executed concurrently. However, action *secure* effectively contributes to the preconditions of *buildShelter*, which means they can be executed sequentially.

A sequence of actions satisfies a goal, if the goal requirements are satisfied for at least the duration of the maintenance period of the goal.

Definition 3.10 (Goal Satisfaction). A sequence of actions π satisfies goal g, denoted $\pi \models g$, iff:

$$\forall i \in [m_s(g), m_e(g)] \cdot s_i \in S(\pi), s_i \models r(g). \tag{4}$$

[5] Recall that in our formalism, it is assumed that the preconditions of an action are maintained until the action completes.

The set of goals satisfied by π is denoted as G_π: $G_\pi = \{g \in G \mid \pi \models g\}$.

A plan is a sequence of actions that fulfills the following criteria: (i) the sequence of actions starts in the initial state; (ii) the preconditions of the actions are fulfilled at all states between the action commencing and completing[6] (sequences of actions only deal with the post conditions); (iii) no action conflicts arise; and (iv) at least one goal is satisfied. These conditions are formulated in the following definition.

Definition 3.11 (Plan). A sequence of actions $\pi = \langle (a_0, 0), \ldots, (a_n, t_{a_n}) \rangle$ is a plan for the normative practical reasoning problem $P = (FL, \Delta, A, G, N)$ iff:

(i) $s_0 = \Delta$,
(ii) $\forall k \in [t_{a_i}, t_{a_i} + d(a_i)) \cdot s_k \models pr(a_i)$,
(iii) $\nexists (a_i, t_{a_i}), (a_j, t_{a_j}) \in \pi t_{a_i} \leq t_{a_j} < t_{a_i} + d(a_i), (a_i, a_j) \in cf_{action}$, and
(iv) $G_\pi \neq \emptyset$.

The set of plans for the agent is denoted as Π.

3.2.1 Normative Properties. Prior to defining norm compliance and violation, below we first define when a norm is activated in a plan.

Definition 3.12 (Activated Norms). A norm $n = (op, a_c, a_s, dl)$ is instantiated and therefore activated in a plan π if its activation condition a_c is executed in the plan. The set of activated norms, denoted N_π is defined as below. Note that $comp_s$ $comp_e$ are calculated based on Definition 3.5.

$$N_\pi = \{(op, a_c, a_s, comp_s, comp_e) \mid (op, a_c, a_s, dl) \in N, (a_c, t_{a_c}) \in \pi\} \tag{5}$$

Now that we know which norms are active, we can determine if they were complied with or violated.

Definition 3.13 (Obligation Compliance). Let π be a plan and also let $n = (o, a_c, a_s, comp_s, comp_e)$ be a norm activated in the plan. We say that π complies with n iff the obliged actions starts during the compliance period:

$$\pi \models n \text{ iff } (a_s, t_{a_s}) \in \pi \text{ s.t. } t_{a_s} \in [comp_s, comp_e) \tag{6}$$

If the action does not start before the compliance period ends the obligation is violated, denoted $\pi \not\models n$.

Definition 3.14 (Prohibition Compliance). Let π be a plan and also let $n = (f, a_c, a_s, comp_s, comp_e)$ be a norm activated in the plan. We say that π complies with n iff the prohibited action does not start during the compliance period:

$$\pi \models (n) \text{ iff } \forall (a_s, t_{a_s}) \in \pi \cdot t_{a_s} \notin [comp_s, comp_e) \tag{7}$$

If the action does start during the compliance period the prohibition is violated, denoted $\pi \not\models n$.

For simplicity, if an active norm has not been violated or complied with during the makespan of the plan, it is considered as violated. So all active norms in a plan are either complied with or violated. The set of norms complied with and violated in plan π are denoted as $N_{cmp(\pi)}$ and $N_{vol(\pi)}$, respectively.

[6]The brackets and parenthesis used in the denotation of time intervals indicate the inclusion and exclusion of interval endpoints, respectively.

3.3 Conflict

In this section we define various types of conflicts within and between goals and norms. Note that since the conflicts are looked at in the context of a plan, we are dealing with the instantiated version of the norms activated in plans.

Goal g_i and g_j are in conflict if their maintenance period overlaps and satisfying them requires bringing about conflicting state of affairs.

Definition 3.15 (Conflicting Goals). Let $P = (FL, \Delta, A, G, N)$ be a the normative practical reasoning problem. The set of conflicting goals, cf_{goal}, is defined as:

$$cf_{goal} = \{(g_i, g_j) \mid \exists\, g_i, g_j \in G \cdot (r(g_i)^+ \cap r(g_j)^-) \cup (r(g_i)^- \cap r(g_j)^+) \neq \emptyset \\ \text{and } [m_s(g_1), m_e(g_1)] \cap [m_s(g_2), m_e(g_2)] \neq \emptyset\}$$

Example 3.16. $runningHospital = \left(\left\{\begin{array}{c} medicsPresent, \\ waterSupplied, \\ medicineSupplied \end{array}\right\}, 5, 8\right)$ and $restrainFlood = \left(\left\{\begin{array}{c} \neg waterSupplied, \\ floodPanelInstalled \end{array}\right\}, 6, 10\right)$, are in conflict because they require the agent to bring about contradictory state of affairs in an overlapping time period.

Next is conflict between goals and norms. An obligation and a goal can be in conflict with respect to a plan if the norm is active during the maintenance period of the goal and the subject of the obligation has postconditions that are contrary to the requirements of the goal. If the norm is to be complied with, a_s's execution should start within the compliance period and end within the following period: $[comp_s + d(a_s), comp_e + d(a_s))$ at which point the postcondition is applied. If the latter period is the subset of maintenance period of goal g, there in no way for the agent to be able to comply with the obligation while satisfying the goal.

Definition 3.17 (Conflicting Obligations and Goals). Let $P = (FL, \Delta, A, G, N)$ be a normative practical reasoning problem. The set of conflicting goals and obligation norms with respect to a plan π, denoted $cf^{\pi}_{goalobl}$, is defined as

$$cf^{\pi}_{goalobl} = \{(g, n) \mid n = (o, a_c, a_s, comp_s, comp_e) \in N_{\pi}; \\ \exists\, r \in r(g) \cdot \neg r \in ps(a_s), \\ [comp_s + d(a_s), comp_e + d(a_s)) \subseteq [m_{s_g}, m_{e_g}]\}$$

Example 3.18. The postcondition of *stopWater* in $n_2 = (o, detectContamination, stopWater, 2)$, namely $\{\neg waterSupplied\}$, contradicts the requirements of goal $runningHospital = \left(\left\{\begin{array}{c} medicsPresent, \\ waterSupplied, \\ medicineSupplied \end{array}\right\}, 5, 8\right)$. Depending on the compliance period of the instantiated version of this norm, *runningHospital* and n_2 can be in conflict in the context of some plans.

We now turn our attention to conflicting prohibition and goals. If a prohibition norm is to be complied with, a_s's execution should not start within compliance period, which makes it impossible for the end state of the action to be within the following period: $[comp_s + d(a_s), comp_e + d(a_s))$. When the postconditions of a_s contribute to the requirements of goal g and that there is some overlap between the maintenance period of goal g, $[m_{s_g}, m_{e_g}]$, and $[comp_s + d(a_s), comp_e + d(a_s))$, the agent cannot comply with the prohibition while satisfying the goal.

Definition 3.19 (Conflicting Prohibitions and Goals). Let $P = (FL, \Delta, A, G, N)$ be a normative practical reasoning problem. The set of conflicting goals and prohibition norms with respect to a plan π, denoted $cf^{\pi}_{goalpro}$, is defined as

$$\begin{aligned} cf^{\pi}_{goalpro} = &\{(g, n) \mid n = (f, a_c, a_s, comp_s, comp_e) \in N_{\pi}; \\ &\exists\, r \in r(g) \cdot r \in ps(a_s), \\ &[comp_s + d(a_s), comp_e + d(a_s)) \cap [m_{s_g}, m_{e_g}] \neq \emptyset\} \end{aligned}$$

Example 3.20. Goal $organiseSurvivorCamp = \left\langle \left\{ \begin{array}{c} areaSecured, \\ shelterBuilt \end{array} \right\}, 15, 15 \right\rangle$ and $n_1 = (f, detectShock, buildShelter, 3)$ can possibly be in conflict in the context of certain plans, because a postcondition of *shelterBuilt*, namely *shelterBuilt*, contributes to the satisfaction of *organiseSurvivorCamp*, however the action is forbidden by the norm.

The entire set of conflicting goals and norms is defined as: $cf^{\pi}_{goalnorm} = cf^{\pi}_{goalobl} \cup cf^{\pi}_{goalpro}$. Next, we define conflicts between norms.

Prior to defining conflict between norms, we note the distinction between direct and indirect conflict. In the former category the conflict arises between norms with contrary deontic operators (e.g.; obligation and prohibition), whereas in the latter the conflict is defined considering the characteristics of the domain the agent operate in [dos Santos et al. 2018]. When norms are action based, as it is the case here, the indirect conflict can occur due to consequences of actions or their causal effects [Aphale et al. 2014; Vasconcelos et al. 2009]. The conflict between obligations (Definition 3.21), that is essentially caused by the pre- and postconditions of actions, falls into the indirect conflict category. This can be contrasted with conflict between obligations and prohibitions (Definition 3.23), which is a direct type of conflict.

Two obligations are in conflict in the context of π if their obliged actions have a concurrency conflict, and one obliged action is in progress during the entire period over which the agent is obliged to execute the other obliged action.

Definition 3.21 (Conflicting Obligations). Let $P = (FL, \Delta, A, G, N)$ be a normative practical reasoning problem. The set of conflicting obligations with respect to a plan π, denoted cf^{π}_{oblobl} is defined as:

$$\begin{aligned} cf^{\pi}_{oblobl} = &\{(n_1, n_2) \mid n_1 = (o, a_c, a_s, comp_s, comp_e), n_2 = (o, b_c, b_s, comp'_s, comp'_e) \in N_{\pi}; \\ &(a_s, b_s) \in cf_{action}; \\ &n_1 \in N_{cmp(\pi)}; \\ &[comp'_s, comp'_e) \subseteq [t_{a_s}, t_{a_s} + d(a_s))\} \end{aligned}$$

Example 3.22. Take $n_2 = (o, detectContamination, stopWater, 2)$ and $n_4 = (o, buildShelter, distributeWater, 5)$. Due to the concurrency conflict between action $stopWater = \left\langle \left\{ \begin{array}{c} contaminationDetected, \\ waterSupplied \end{array} \right\}, \{\neg waterSupplied\}, 1 \right\rangle$, and action $distributeWater = \langle \{\}, \{waterSupplied\}, 1 \rangle$ it is possible that in some plans they create conflict.

An obligation and a prohibition are in conflict in the context of π if the prohibition forbids the agent to execute an action during the entire period over which the obligation obliges the agent to take the same action.

Definition 3.23 (Conflicting Obligations and Prohibitions). Let $P = (FL, \Delta, A, G, N)$ be a normative practical reasoning problem. The set of conflicting obligations and prohibitions with respect to a plan π, denoted cf^{π}_{oblpro} is defined as:

$$cf^{\pi}_{oblobl} = \{(n_1, n_2) \mid n_1 = (o, a_c, a_s, comp_s, comp_e), \\ n_2 = (f, a_c, a_s, comp'_s, comp'_e) \in N_\pi; \\ [comp_s, comp_e) \subseteq [comp'_s, comp'_e)\}$$

Example 3.24. $n_2 = (o, detectContamination, stopWater, 2)$ and $n_3 = (f, buildShelter, stopWater, 3)$ can be in conflict in some plans as they require and forbid stopping water supply.

The two sets cf^{π}_{oblobl} and cf^{π}_{oblpro} constitute the set of conflicting norms: $cf^{\pi}_{norm} = cf^{\pi}_{oblobl} \cup cf^{\pi}_{oblpro}$.

4 IDENTIFYING THE BEST PLAN

In the previous section we described a formal model for an agent that is capable of devising plans for achieving multiple goals while complying with norms. However, conflicts often make it impossible for the agent to satisfy all goals while complying with all the norms that a plan may trigger. The agent therefore needs to reason about all these conflicts while taking its priorities, expressed in terms of preferences between goals and norms, into account to decide on the best plan to execute. In this section, we consider how formal argumentation can enable an agent to select appropriate plans to execute, taking all these factors into account.

Arguing over the appropriateness of a plan involves putting forward the plan as a proposal and letting the agent question the justifiability of the plan proposal by investigating why a certain goal is not satisfied in the proposed plan, or why a certain norm is violated. The evaluation of argumentation frameworks for the plan proposals results in identifying justified plans. The justified plans are further refined in a search for the best plan, by comparing the quality (i.e., preferences) and quantity (i.e., numbers) of goals satisfied and norms violated in these plans.

4.1 Plan Proposal Argumentation Frameworks

An argumentation framework (AF) is defined as below.

Definition 4.1. [Argumentation Framework [Dung 1995]] An argumentation framework is a pair $AF = \langle Arg, Att \rangle$, where Arg is a finite set of arguments and Att is an attack relation between arguments: $Att \subseteq Arg \times Arg$.

Arguments represent defeasible logical inferences, while attacks show the inconsistency between arguments. In scheme-based approaches arguments are expressed in natural language as defeasible rules. If the facts in the premise of an argument scheme hold the argument scheme will be instantiated into an argument. A set of critical questions is associated with each scheme, identifying how the instantiated arguments can be attacked.

Argumentation semantics [Dung 1995] are means of evaluating arguments in an AF. Caminada [Caminada 2006] provides an intuitive way to identify the status of arguments with respect to various semantics through labellings: an argument is labelled *in*, *out* and *undec*, if it is acceptable, rejected or undecided, respectively, under a certain semantics. In a complete labelling an argument is labelled *in* iff all its attackers are labelled *out*, and is labelled *out* iff there exists an attacker for it that is labelled *in*.

In deciding what semantics to use, the way conflicts are dealt with by different semantics is particularly important. In what follows, we first explain how arguments, and the attacks between them, are identified when reasoning about a plan proposal. We then justify why *credulous preferred* is the suitable semantics for evaluating plan proposal AFs.

4.1.1 Generating Arguments. Below, we introduce a set of argument schemes and critical questions to reason about a plan proposal with respect to the goals it satisfies and the norms it complies with or violates. Our schemes provide arguments for why a plan should – or should not – be executed.

When instantiated, the most basic scheme we consider involves constructing an argument for each possible plan that exists. This scheme is inspired by Oren's scheme [Oren 2013] for a sequence of actions and Atkinson's scheme for plans in BDI agents [Atkinson 2005], and is referred to as a plan argument scheme.

Definition 4.2 (Plan Argument Scheme Arg_π). A plan argument claims that a proposed sequence of actions should be executed because it satisfies a set of goals and complies with a set of norms while violating some other norms:

- In the initial state Δ
- The agent should execute sequence of actions π
- Which will satisfy a set of goals G_π and complies with a set of norms $N_{cmp(\pi)}$.

The next scheme results in constructing an argument for each goal that is feasible (i.e., satisfied in at least one plan). If there is no plan to satisfy a goal, a rational agent should not adopt that goal or try to justify its adoption. Goal arguments are therefore only constructed for feasible goals. A goal argument is used to explore why a goal is not satisfied in a plan, or to address the conflict between two goals or a goal and a norm.

Definition 4.3 (Goal Argument Scheme Arg_g). A goal argument claims that a feasible goal should be satisfied:

- Goal g is feasible for the agent
- Therefore, satisfying g is required.

The set of goal arguments is denoted as Arg_G.

Finally, we consider an argument scheme that creates arguments for each activated norm within a plan. Such norm arguments are used to explore why a norm is violated in a plan. It is also used to address the conflict between two norms or a goal and a norm.

Definition 4.4 (Norm Argument Scheme Arg_n). A norm argument claims that an activated norm should be complied with:

- n is an activated norm imposed on the agent in plan π
- Therefore, complying with n is required in π.

The set of norm arguments for a plan π is denoted as Arg_{N_π}.

4.1.2 Critical Questions and Interactions between Arguments. Arguments may challenge each other either by having contradictory conclusions, or by expressing inconsistencies in another way. Such inconsistencies are captured through critical questions, which attack the use of an argument by challenging or rejecting it based on the way it was instantiated from an argument scheme. We now describe the critical questions associated with each argument scheme.

Critical Questions for the Plan Argument Scheme

CQ1: A plan should not be followed if it does not achieve a goal. This critical question, informally asking whether *a goal argument is not achieved by the plan*, results in an asymmetric attack from the goal argument to the plan argument. Formally, this occurs when the following condition holds.

$$\forall Arg_g \in Arg_G \text{ if } \pi \not\models g \text{ then } (Arg_g, Arg_\pi) \in Att$$

CQ2: The violation of a norm by a plan provides a reason why the plan should not be followed, resulting in an attack from the norm's argument to the plan argument. This critical question asks *whether a norm is violated by the plan*, and is formally encoded as follows.

$$\forall Arg_n \in Arg_{N_\pi} \text{ if } \pi \not\models n \text{ then } (Arg_n, Arg_\pi) \in Att$$

Critical Questions for the Goal Argument Scheme

CQ3: Mutually exclusive goals provide reasons why one or the other goal should not be pursued, resulting in a symmetric attack between goal arguments. This critical question asks *whether some other goal can be achieved if this goal is not*, leading to the following formal definition.

$$\forall Arg_g, Arg_{g'} \in Arg_G \text{ if } (g, g') \in cf_{goal} \text{ then } (Arg_g, Arg_{g'}) \in Att$$

CQ4: What norm arguments might attack Arg_g? A conflicting goal and norm force an agent to choose between them resulting in a symmetric attack between them. This critical question asks if *there is some norm whose compliance prevents the goal from being achieved.*

$$\forall Arg_g \in Arg_G, Arg_n \in Arg_{N_\pi} \text{ if } (g, n) \in cf^{\pi}_{goalnorm} \text{ then } (Arg_g, Arg_n) \in Att$$

Critical Questions for the Norm Argument Scheme

CQ4: This question is the dual of the previous one (CQ4 for goal argument schemes), stating that a norm might not be complied with if it stands in the way of achieving a goal.

$$\forall Arg_g \in Arg_G, Arg_n \in Arg_{N_\pi} \text{ if } (n, g) \in cf^{\pi}_{goalnorm} \text{ then } (Arg_n, Arg_g) \in Att$$

CQ5: Two conflicting norms force an agent to choose between them. This can be informally specified as the critical question of *whether there is some other norm that is in conflict with this norm*, and results in a symmetric attack between the two norms, which is formally represented as follows.

$$\forall Arg_n, Arg_{n'} \in Arg_{N_\pi} \text{ if } (n, n') \in cf^{\pi}_{norm} \text{ then } (Arg_n, Arg_{n'}) \in Att$$

4.1.3 Preference Relation between Arguments. Generating arguments and applying the critical questions above results in a set of arguments and attacks between them. However, an agent may prioritise certain goals and norms over others, and these priorities can be reflected in the agent's reasoning by encoding them as preferences. Such preferences allow one to distinguish an attack from a *defeat* (i.e., a successful attack [Amgoud and Cayrol 2002]).

Definition 4.5 (Preference between Goals and Norms). We define $\succeq^{gn}$ as a partial preorder on $G \cup N$, where $\alpha, \beta \in G \cup N$ denotes that satisfying goal α (or complying with norm α) is at least as preferred as satisfying goal β (or complying with norm β). Symbol $\succ^{gn}$ denotes the strict relation corresponding to $\succeq^{gn}$: $(\alpha, \beta) \in \succ^{gn}$ iff $(\alpha, \beta) \in \succeq^{gn}$ and $(\beta, \alpha) \notin \succeq^{gn}$. $(\alpha, \beta) \in \sim^{gn}$ iff $(\alpha, \beta) \in \succeq^{gn}$ and $(\beta, \alpha) \in \succeq^{gn}$.

The preferences between the goal and norm arguments result from the preference between these goals and norms: $(Arg_\alpha, Arg_\beta) \in \geq$ iff $(\alpha, \beta) \in \succeq^{gn}$.

Each possible plan (a *plan proposal*) has an AF associated with it, consisting of the argument for the plan, a set of arguments for goals, and arguments for norms that are activated in that plan. Although the set of goal arguments in AFs for plan proposals remain the same across the AFs, the set of norm arguments differs between AFs depending on the norms that are activated by the plan proposed within the AF.

Definition 4.6 (Plan Proposal AF). The AF for plan proposal π is denoted as $AF_\pi = \langle Arg, Def \rangle$, where $Arg = Arg_\pi \cup Arg_G \cup Arg_{N_\pi}$ and $\forall Arg_\alpha, Arg_\beta \in Arg, (Arg_\alpha, Arg_\beta) \in Def$ iff $(Arg_\alpha, Arg_\beta) \in Att_{CQ1-5}$ and $(Arg_\beta, Arg_\alpha) \notin \succ$.

We have described how a set of argument frameworks can be constructed from a set of norms, goals and plans. In the next section, we describe how these argument frameworks can be evaluated.

4.2 Evaluating the Plan Proposal Argumentation Frameworks

As pointed out in Section 4.1 evaluating AFs is done using argumentation semantics [Dung 1995]. *Credulous* semantics preserves choices and produces multiple alternatives in the case of unresolvable conflict between arguments, whereas *sceptical* semantics rejects both arguments in an unresolvable conflict. Works [Broersen et al. 2002; Oren 2013; Prakken 2006; Thomason 2000] in the field of argumentation-based practical reasoning and decision-making are unanimous that reasoning about and toward actions has to be credulous. If the conflict between goals for example is unresolvable then what we want is to have choices between them rather than rejecting both of them. We refer the readers to [Caminada 2006] and [Prakken 2006] for the philosophical and pragmatic foundation of credulous inference for practical reasoning.

Taking membership of the preferred extension as a justified viewpoint that the agent can adopt, justified plans are defined as those labelled *in* by a preferred labelling of their associated *AF* (i.e.; if π is labelled *in* in AF_π). In such a situation, one can conclude that the plan is justifiable with respect to the agent's set of goals and set of norms activated in that plan, along with conflicts and preferences between the set of goals and norms.

Definition 4.7 (Preferred Labellings). A complete labelling is called a preferred labelling, iff its set of in-labelled arguments is maximal (with respect to set inclusion), or equivalently, iff its set of out-labelled arguments is maximal (with respect to set inclusion).

Definition 4.8 (Justified Plans). Plan π is *justified* if Arg_π is labelled *in* by at least one preferred labelling for AF_π: $\exists\ \mathcal{L}_{pr}$ s.t. $Arg_\pi \in in(\mathcal{L})$.

Although all the justified plans are internally coherent and defendable by the agent, there could be further criteria that allows one to decide that some plan is more preferred, or better, than some other plan. Identifying an ordering between justified plans has been treated differently. Some works (e.g., [Amgoud et al. 2008b]) do not distinguish between justified plans and regard them as "as good as" each other. Therefore, all the justified plans are the best plans. Simply maximising the number of achieved desires is the basis of comparison of justified options in [Hulstijn and van der Torre 2004], while [Rahwan and Amgoud 2006] uses the utility of plans (i.e., the worth of desires and the cost of resources to achieve them) to find the best plan out of the justified ones. In this work, what is available to the agent is a partial preference order over goals and norms. We therefore, use the established set ordering technique [Amgoud and Vesic 2014; Caminada et al. 2014b; Prakken and Sartor 1997] known as the *democratic ordering* for further refinement of justified plans by considering the combination of goals satisfied and norms violated in these plans. Since preferences over goals and norms are partial, comparing two plans based on the democratic ordering is not always possible. Therefore, in the absence of such preference information, the best plan is defined as the one that satisfies the most goals while violating fewest norms. We first give a formal account of the democratic ordering and then define the *goal-dominant* and *norm-dominant* plans, based on which a *better than* relation between plans is defined.

Definition 4.9 (Democratic Ordering). Let S_i and S_j be two sets of objects. According to the democratic ordering (denoted $\trianglerighteq$) $(S_i, S_j) \in \trianglerighteq$ iff $\forall \beta \in S_j \setminus S_i, \exists \alpha \in S_i \setminus S_j$ s.t. $(\alpha, \beta) \in \succ$. As usual $(S_i, S_j) \in \triangleright$ iff $(S_i, S_j) \in \trianglerighteq$ and $(S_i, S_j) \notin \trianglerighteq$.

The democratic ordering is reflexive and transitive.[7]

Definition 4.10 (Goal Dominance: Democratic Ordering). Let G_{π_i} and G_{π_j} be the sets of goals satisfied in plan π_i and π_j. According to the democratic ordering $(G_{\pi_i}, G_{\pi_j}) \in \trianglerighteq_G$ iff $\forall g' \in G_{\pi_j} \setminus G_{\pi_i}, \exists g \in G_{\pi_i} \setminus G_{\pi_j}$ s.t. $(g, g') \in \succ^{gn}$.

Let $\succ^g$ be the ordering on G induced by $\succ^{gn}$.

Theorem 4.11. *$\trianglerighteq_G$ is a total preorder on $P(G)$ iff $\succ^g$ is a total preorder.*[8]

See Appendix B for the proof of this theorem and the rest of propositions and properties in this Section.

Given theorem 4.11, we must identify when $\trianglerighteq_G$ (Definition 4.10) is a total preorder over $G_\Pi = \{G_{\pi_1}, G_{\pi_2}, \cdots, G_{\pi_n}\}$, where G_{π_i} is the set of goals satisfied by plan π_i.

Corollary 4.12. *$\trianglerighteq_G$ is a total preorder on G_Π if $\succ^g$ is total.*

The above corollary follows immediately from Theorem 4.11 because $G_\Pi \subseteq P(G)$. However, note that having a total preorder on $\trianglerighteq_G$, does not necessarily mean we need to have a total order on G. Here is a counter example.

Example 4.13. Let $G = \{g_1, g_2, g_3\}$ and $\succ^g = \{(g_2, g_3)\}$. Also let $G_\Pi = \{G_{\pi_1}, G_{\pi_2}\}$, where $G_{\pi_1} = \{g_1, g_2\}$ and $G_{\pi_2} = \{g_1, g_3\}$. It is clear that $(G_{\pi_1}, G_{\pi_2}) \in \trianglerighteq_G$. Therefore, we have a total preorder on G_Π while $\succ^g$ is not a total order on G.

We are now in a position to define *goal-dominance*. Plan π_i *goal-dominates* plan π_j in two ways. First, if for every goal satisfied in π_j which is not satisfied in π_i, there is at least one preferred goal satisfied in π_i which is not satisfied in π_j. Second, if there is insufficient preference information available, one plan goal-dominates another if the former satisfies more goals than the latter.

Definition 4.14 (Goal-dominance). Plan π_i goal-dominates π_j, denoted as $(\pi_i, \pi_j) \in \geq_G$

(1) If $\trianglerighteq_G$ is a total preorder on G_Π and $(G_{\pi_i}, G_{\pi_j}) \in \trianglerighteq_G$;

(2) If $\trianglerighteq_G$ is not a total preorder on G_Π and $|G_{\pi_i}| \geq |G_{\pi_j}|$.

$>_G$ is the strict relation associated with $\geq_G$. $(\pi_i, \pi_j) \in \sim_G$ iff $(\pi_j, \pi_i) \in \geq_G$ and $(\pi_i, \pi_j) \in \geq_G$.

It is straight forward to see $>_G$ is irreflexive, antisymmetric and transitive, while $\sim_G$ is reflexive, symmetric and transitive.

Note that – as done by [Amgoud and Prade 2009] – goal-dominance could equivalently be defined in argument level (i.e.; in terms of preferences over *justified* goal arguments, and the number of justified goal arguments appearing within the plan (c.f., Properties 4 and 5 on page 16)).

Similar to Definition 4.14, norm dominance is defined based on the democratic ordering and number of violations. Plan π_i norm-dominates plan π_j if for every norm violated in π_i that is not violated in π_j, there is at least one stronger norm violated in π_j that is not violated in π_i.

Definition 4.15 (Norm Dominance: Democratic Ordering). Let $N_{vol(\pi_i)}$ and $N_{vol(\pi_j)}$ be the sets of norms violated in plan π_i and π_j. According to the democratic ordering $(N_{vol(\pi_i)}, N_{vol(\pi_j)}) \in \trianglerighteq_N$ iff $\forall n \in N_{vol(\pi_i)} \setminus N_{vol(\pi_j)}, \exists n' \in N_{vol(\pi_j)} \setminus N_{vol(\pi_i)}$ s.t. $(n', n) \in \succ^{gn}$.

Let $N_{vol(\Pi)} = \{N_{vol(\pi_1)}, N_{vol(\pi_2)}, \cdots, N_{vol(\pi_n)}\}$, where as before $N_{vol(\pi_i)}$ is the set of norms violated in plan i.

[7] See [Amgoud and Vesic 2014] for proof.

[8] $P(G)$ is the power set of G.

Definition 4.16 (Norm-dominance). Plan π_i norm-dominates π_j denoted as $(\pi_i, \pi_j) \in \geq_N$

(1) If $\trianglerighteq_N$ is a total preorder on $N_{vol(\Pi)}$ and $(N_{vol(\pi_i)}, N_{vol(\pi_j)}) \in \trianglerighteq_N$;

(2) If $\trianglerighteq_N$ is not a total preorder on $N_{vol(\Pi)}$ and $|N_{vol(\pi_j)}| \geq |N_{vol(\pi_i)}|$.

$>_N$ is the strict relation associated with $\geq_N$. $(\pi_i, \pi_j) \in \sim_N$ iff $(\pi_j, \pi_i) \in \geq_N$ and $(\pi_i, \pi_j) \in \geq_N$.

It is straight forward to see that $>_N$ is irreflexive, antisymmetric and transitive, while $\sim_N$ is reflexive, symmetric and transitive.

Goal dominance and norm dominance can be defined and combined in various ways to provide the basis for plan comparison. For examples, a norm-dominant plan can be defined as a plan with the highest return on norm compliance, rather than minimum violation, or in combining these factors norm dominance can be given priority over norm dominance. The general idea, however will be the same: we favour plans that are goal-dominant and norm-dominant. In what follows, we give priority to dominance of goals over norms. The dominance of norms can be given priority over the dominance of goals by swapping the order of conditions 2 and 3 in the following definition.

Definition 4.17 (Plan Comparison). Plan π_i is better than π_j, denoted $(\pi_i, \pi_j) \in >_\pi$, iff:

(1) π_i is justified and π_j is not; or

(2) π_i and π_j are both justified and $(\pi_i, \pi_j) \in >_G$; or

(3) π_i and π_j are both justified and $(\pi_i, \pi_j) \in \sim_G$ but $(\pi_i, \pi_j) \in >_N$.

Plan π_i is as good as π_j, denoted $(\pi_i, \pi_j) \in \sim_\pi$, iff $(\pi_i, \pi_j) \notin >_\pi$ and $(\pi_j, \pi_i) \notin >_\pi$.

Proposition 4.18. *$>_\pi$ is irreflexive.*

Proposition 4.19. *$>_\pi$ is antisymmetric.*

Proposition 4.20. *$>_\pi$ is transitive.*

Proposition 4.21. *$\sim_\pi$ is an equivalence relation on Π.*

Definition 4.22 (Plan Equivalence Classes). Given $\pi \in \Pi$, let $[\pi_i]$ denote the equivalence class to which π_i belongs. $([\pi_i], [\pi_j]) \in \geq$ iff $(\pi_i, \pi_j) \in >_\pi$ or $(\pi_i, \pi_j) \in \sim_\pi$.

Proposition 4.23. *$\geq$ is a total order on Π.*

Definition 4.24 (Best Plan). Plan π_i is the best plan for the agent to execute iff

- π_i is justified, and
- $\nexists \pi_j$ such that $([\pi_j], [\pi_i]) \in \geq$.

Based on this definition, there might be more than one plan identified as a best plan. In the case of a single agent, any of these plans can be chosen at random.[9] For argumentation-based plan construction and plan selection in a multi-agent setting, see [Ferrando and Onaindia 2017] and [Belesiotis et al. 2010], respectively. In these work, agents participate in iterated dialogues by exchanging arguments about different plan proposals, until they reach an agreement.

Prior to give a comprehensive example (Section 5) on constructing plan proposal AF, evaluating them and identifying the best plan based on them, we investigate the formal properties of our framework in the next section.

[9] Additional refinement criteria, such as selecting the shortest plan, could be introduced to filter the best plan set further.

4.3 Properties of the Framework

We now consider the properties of our framework with regards to *rationality postulates* [Caminada and Amgoud 2007], which identify desired properties of argumentation systems (see Properties 1 to 3). Compliance with rationality postulates demonstrates that our argumentation system satisfies the fundamental properties required of an argumentation system. We also investigate the properties of the best plan(s) and the preferred extensions that include it (see Properties 4 to 7). In addition, we show the satisfiability of the Maximal Elements Postulate [Amgoud and Vesic 2014] (see property 8). The intuition behind proving this property is to show that our framework privileges the maximal elements (i.e., strongest arguments). That is, the strongest arguments in an AF, when conflict-free, are all labelled *in* by the preferred labelling that labels the plan argument *in*. The last property (Property 9) is inspired by Brewka and Eiters' Principle I [Brewka and Eiter 2000], which is a principle for sound extension-based reasoning with preferences. Basically, the principle expresses that if two extensions only differ in one argument, depending on the preference between the distinct argument in each, the viewpoint presented by one can be given priority over the other.

Property 1. *Rationality Postulate, Closure: The conclusions of any extension (in labelled arguments) are closed under strict rules.*

Property 2. *Rationality Postulate, Direct Consistency: The conclusions of any extension are consistent.*

Property 3. *Rationality Postulate, Indirect Consistency: The closure under strict rules of the conclusions of any extension is consistent.*

Property 4. *If a plan argument is labelled in by preferred labelling $\mathcal{L}$, the arguments representing all the goals that it does not satisfy and the norms it violates are labelled out by $\mathcal{L}$ and vice versa:*

$$Arg_\pi \in in(\mathcal{L}) \Leftrightarrow \bigcup_{g \in G \setminus G_\pi} Arg_g \cup \bigcup_{n \in N_{vol(\pi)}} Arg_n \subseteq out(\mathcal{L})$$

Property 5. *If a plan argument is labelled in by preferred labelling $\mathcal{L}$, the arguments representing all the goals that it satisfies and norms it complies with are also labelled in by $\mathcal{L}$:*

$$Arg_\pi \in in(\mathcal{L}) \Rightarrow \bigcup_{g \in G_\pi} Arg_g \cup \bigcup_{n \in N_{cmp(\pi)}} Arg_n \subseteq in(\mathcal{L})$$

Note that from $\bigcup_{g \in G_\pi} Arg_g \cup \bigcup_{n \in N_{cmp(\pi)}} Arg_n \subseteq in(\mathcal{L})$ one cannot conclude that $Arg_\pi \in in(\mathcal{L})$, as there might be justified goals or norms not satisfied or complied with by the plan.

Property 6. *There is no more than one preferred labelling in which $Arg_\pi \in in(\mathcal{L})$.*

Property 7. *If $Arg_\pi \in in(\mathcal{L})$ then $\mathcal{L}$ is a stable labelling.*

Property 8. *Let $\geq^{gn}$ be a total preorder on $G \cup N$ and therefore $\geq$ be a total preorder on goal and norm arguments. If $Arg_\pi \in in(\mathcal{L})$, and the set of arguments for the most preferred goals and norms, $Pref(Arg)$, is conflict free then all arguments belong to $Pref(Arg)$ are labelled in by $\mathcal{L}$.*

Property 9. *Assume that plan π_1 and π_2 are both justified (i.e., $Arg_{\pi_1} \in in(\mathcal{L}_{AF_{\pi_1}})$ and $Arg_{\pi_2} \in in(\mathcal{L}_{AF_{\pi_2}})$). Let the preferred extensions that contain Arg_{π_1} and Arg_{π_2} be E_1 and E_2, respectively such that (i) $E_1 \setminus \{Arg_{\pi_1}\} = E_0 \cup \{Arg_\alpha\}$ and $E_2 \setminus \{Arg_{\pi_2}\} = E_0 \cup \{Arg_\beta\}$; (ii) $Arg_\alpha, Arg_\beta \notin E_0$; and (iii) $(Arg_\alpha, Arg_\beta) \in \succ$. It holds that plan π_1 is always better than plan π_2 (i.e., $(\pi_1, \pi_2) \in >_\pi$) except when Arg_α is a norm argument and Arg_β is a goal argument.*

5 ILLUSTRATIVE SCENARIO: PART II

To illustrate our approach we proposed a scenario in Section 2. The full formulation of this scenario is as follows. Let $P = (FL, \Delta, A, G, N)$ be the normative practical reasoning problem for the Disaster Scenario such that:

- $FL = \left\{ \begin{array}{c} shockDetected, contaminationDetected, waterSupplied, \\ areaSecured, evacuated, shockDetected, \\ shelterBuilt, populated, wounded, \\ earthquakeDetected, medicineSupplied, \\ noAccess, medicsPresent \end{array} \right\}$
- $\Delta = \left\{ \begin{array}{c} earthquakeDetected, medicsPresent, \\ wounded, populated, waterSupplied \end{array} \right\}$
- $A = \left\{ \begin{array}{c} detectShock, detectContamination, stopWater, \\ buildShelter, evacuate, getMedicine, secure \end{array} \right\}$ where

  $detectShock = \langle \{\}, \{shockDetected\}, 1 \rangle$. $detectContamination = \langle \{\}, \{contaminationDetected\}, 1 \rangle$.

  $stopWater = \left\langle \left\{ \begin{array}{c} contaminationDetected, \\ waterSupplied \end{array} \right\}, \{\neg waterSupplied\}, 1 \right\rangle$

  $buildShelter = \left\langle \left\{ \begin{array}{c} areaSecured, \\ evacuated, \end{array} \right\}, \left\{ \begin{array}{c} shelterBuilt, \\ \neg evacuated \end{array} \right\}, 2 \right\rangle$, $evacuate = \left\langle \left\{ \begin{array}{c} shockDetected, \\ populated \end{array} \right\}, \left\{ \begin{array}{c} evacuated, \\ \neg populated \end{array} \right\}, 5 \right\rangle$

  $getMedicine = \left\langle \left\{ \begin{array}{c} earthquakeDetected, \\ wounded \end{array} \right\}, \{medicine\}, 3 \right\rangle$, $secure = \left\langle \{evacuated\}, \left\{ \begin{array}{c} areaSecured, \\ noAccess \end{array} \right\}, 5 \right\rangle$
- $G = \{runningHospital, organiseSurvivorCamp\}$, where:

  $runningHospital = \left\langle \left\{ \begin{array}{c} medicsPresent, \\ waterSupplied, \\ medicineSupplied \end{array} \right\}, 5, 8 \right\rangle$, and $organiseSurvivorCamp = \left\langle \left\{ \begin{array}{c} areaSecured, \\ shelterBuilt \end{array} \right\}, 15, 15 \right\rangle$.
- $N = \{n_1, n_2, n_3\}$, where: $n_1 = (f, detectShock, buildShelter, 3)$, $n_2 = (o, detectContamination, stopWater, 2)$, and $n_3 = (f, buildShelter, stopWater, 3)$.

Since the requirements of the two goals are not inconsistent, the set of potential goal-goal conflict is empty: $cf_{goal} = \emptyset$. The requirements of *runningHospital* is in conflict with the postconditions of *stopWater* that is the subject of obligation n_2, because the former requires *waterSupplied*, while the latter brings about $\neg$*waterSupplied*. Goal *organiseSurvivorCamp* does not conflict with n_2, thus the set of potential goal obligation conflict is: $cf^{\pi}_{goalobl} = \{(runningHospital, n_2)\}$. Next is the conflicts between goals and prohibitions. *organiseSurvivorsCamp* can conflict with n_1, becasue the goal requires *shelterBuild*, but n_1 forbids action *buildShelter*, the postcondition for which is *shelterBuild*. *runningHospital* and n_3 can potentially be in conflict in the same fashion. Therefore, the set of potential conflict between goals and prohibitions is: $cf^{\pi}_{goalpro} = \{(organiseSurvivorsCamp, n_1), (runningHospital, n_3)\}$. Since there is a single obligation the set cf^{π}_{oblobl} is empty. Last is the conflict between obligation and prohibition. Since n_2 obliges the agent to take action *stopWater*, while n_3 forbids this action, they can potentially conflict: $cf^{\pi}_{oblpro} = \{(n_2, n_3)\}$. In what follows we show plans in which due to different sequencing of actions, some of these potential conflicts have occurred, and some others have not.

This planning problem is implemented using Answer Set Programming (ASP) [Gelfond and Lifschitz 1988], which is a declarative programming paradigm using logic programs under Answer Set semantics. In this paradigm the user

provides a description of a problem and ASP works out how to solve the problem by returning answer sets corresponding to problem solutions. Since implementation is not the focus of current article, we refer the readers to [Shams et al. 2017] for details of mapping the planning problem to ASP and getting the plans in terms of answer sets.

After generating the plans, the agent has to decide on the best one to follow using argumentation. Let us now consider some plans for this scenario to show how the agent construct their associated AF to check the justifiability of the plans. Note that, in order to make studying plans easier in terms of (i) their sequence of actions, (ii) the fluents that hold in each state, (iii) goal satisfaction and norm compliance/violation, we visualise their associated answer sets in Appendix A.

Example 5.1. Let $\pi_1, \pi_2, \pi_3, \pi_4 \in \Pi$ be four plans for the agent as follows:

$$\pi_1 = \{(getMedicine, 0), (detectShock, 1), (evacuate, 2), (secure, 7), (buildShelter, 12), (detectContamination, 14), (stopWater, 15)\}$$

$$\pi_2 = \{(getMedicine, 0), (detectShock, 2), (evacuate, 3), (detectContamination, 5), (secure, 8), (buildShelter, 13), (stopWater, 15)\}$$

$$\pi_3 = \{(detectShock, 0), (evacuate, 1), (secure, 6), (detectContamination, 7), (stopWater, 8), (buildShelter, 11), (getMedicine, 13)\}$$

$$\pi_4 = \{(detectShock, 0), (evacuate, 1), (secure, 6), (detectContamination, 9), (buildShelter, 11), (getMedicine, 12), (stopWater, 15)\}$$

Plan π_1 and π_2 satisfy both goals *runningHospital* and *organiseSurvivorsCamp*, while π_3 and π_4 only satisfy goal *organiseSurvivorsCamp*. Since actions *detectShock*, *detectContamination*, and *buildShelter*, that are the activation condition of norms n_1, n_2, and n_3, respectively, are executed in all four plans, the set of activated norms in all plans includes an instantiated version of these norms[10]. The set of instantiated norms complied with or violated varies across plans as stated below.

- $\pi_1 \models \{runningHospital, organiseSurvivorsCamp\}$,
  $N_{\pi_1} = \{n_{11}, n_{21}, n_{31}\}$, $N_{cmp(\pi_1)} = \{n_{11}, n_{21}\}$, $N_{vol(\pi_1)} = \{n_{31}\}$
- $\pi_2 \models \{runningHospital, organiseSurvivorsCamp\}$,
  $N_{\pi_2} = \{n_{12}, n_{22}, n_{32}\}$, $N_{cmp(\pi_2)} = \{n_{12}\}$, $N_{vol(\pi_2)} = \{n_{22}, n_{32}\}$
- $\pi_3 \models \{organiseSurvivorsCamp\}$,
  $N_{\pi_3} = \{n_{13}, n_{23}, n_{33}\}$, $N_{cmp(\pi_3)} = \{n_{13}, n_{23}, n_{33}\}$, $N_{vol(\pi_3)} = \{\}$
- $\pi_4 \models \{organiseSurvivorsCamp\}$,
  $N_{\pi_4} = \{n_{14}, n_{24}, n_{34}\}$, $N_{cmp(\pi_4)} = \{n_{14}\}$, $N_{vol(\pi_4)} = \{n_{24}, n_{34}\}$

Figures 1-4 show[11] the AF associated with each of these plans in absence of any preference information. Arrows, representing attacks, are annotated with the CQ that causes the attack. Essentially, the attacks are characterising the

[10] n_{xy} is the instantiated version of norm x in plan y. For example, n_{32} refers to the instantiated version of norm n_3 in π_2, while n'_{32} could be another instantiation of the same norm in the same plan.
[11] *run* and *org* stand for *runningHospital* and *organiseSurvivorsCamp*. Also Pr. ex. is abbreviation for Preferred Extension.

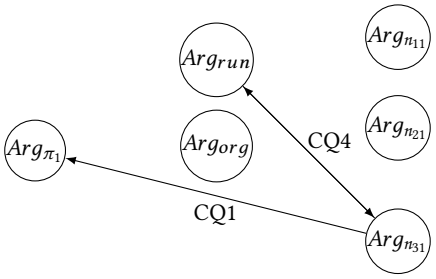

Fig. 1. AF Associated with Plan π_1
Pr. Ex. 1 =$\{Arg_{\pi_1}, Arg_{run}, Arg_{org}, Arg_{n_{11}}, Arg_{n_{21}}\}$
Pr. Ex. 2 =$\{Arg_{org}, Arg_{n_{11}}, Arg_{n_{21}}, Arg_{n_{21}}\}$

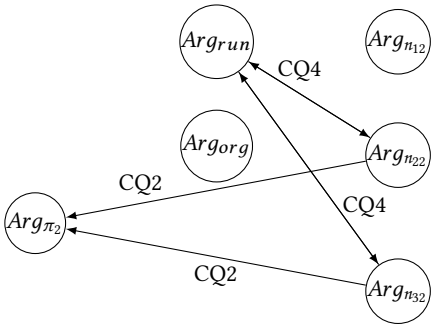

Fig. 2. AF Associated with Plan π_2
Pr. Ex. 1 =$\{Arg_{\pi_2}, Arg_{org}, Arg_{run}, Arg_{n_{12}}\}$
Pr. Ex. 2 =$\{Arg_{org}, Arg_{n_{12}}, Arg_{n_{22}}, Arg_{n_{32}}\}$

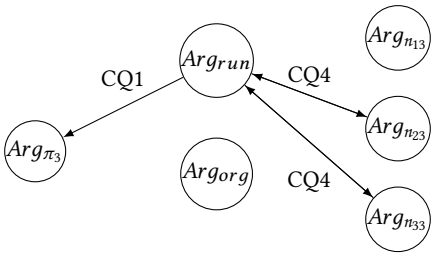

Fig. 3. AF Associated with Plan π_3
Pr. Ex. 1 =$\{Arg_{\pi_3}, Arg_{run}, Arg_{org}, Arg_{n_{13}}\}$
Pr. Ex. 2 =$\{Arg_{org}, Arg_{n_{13}}, Arg_{n_{23}}, Arg_{n_{33}}\}$

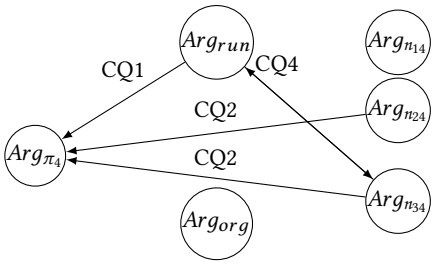

Fig. 4. AF Associated with Plan π_4
Pr. Ex. 1 =$\{Arg_{run}, Arg_{org}, Arg_{n_{14}}, Arg_{n_{24}}\}$
Pr. Ex. 2 =$\{Arg_{org}, Arg_{n_{14}}, Arg_{n_{24}}, Arg_{n_{34}}\}$

conflict that is detected in the plan (i.e.; conflict detection phase). The conflicts are resolved by applying preferred semantics that give rise to extensions listed below each figure (i.e.; conflict resolution phase). For example, if a conflict between argument for norm A and norm B is detected, there will be two preferred extensions, one of which resolves the conflict by recommending complying with norm A and violating B, while the other extension favours complying with B and violating A.

In Figure 1, Arg_{π_1} represents the plan argument that is under consideration in this framework. Since both goals *runningHospital* and *organiseSurvivorsCamp* are satisfied in this plan, arguments Arg_{run} and Arg_{org} do not attack Arg_{π}. On the hand, since norm n_{31} is violated in this plan, $Arg_{n_{31}}$ attacks Arg_{π}, which provides a way to enquire why the plan has violated this norm. The other two norms are complied with in this plan, $Arg_{n_{11}}$ and $Arg_{n_{21}}$, thus do not attack Arg_{π}. The attack between Arg_{run} and $Arg_{n_{31}}$ is due to their conflict and it provides a way to enquire why n_{31} was violated in favour of satisfying goal *runningHospital*. Here, there are no more conflicts between goals and norms, and hence no more attacks in the AF. AFs for plans π_2–π_4, are constructed in the same manner. Notice that depending on the actions executed in each plan and their sequence, some arguments attack each other only in some AFs. For example Arg_{n_2} and Arg_{run} attack each other in π_2 and π_3, but not in π_1 and π_4.

Example 5.2. Consider plans π_1 to π_4 from previous example and assume there are no preferences available to the agent. The first condition of the best plan is justifiability with respect to preferred semantics. Looking into the preferred

extensions of each framework in Figures 1 to 4, we can see that apart from π_4, the plan arguments for the other three plans is acceptable under preferred semantics (the plan argument appears in at least one extension, which is equivalent to being labelled *in* by at least one preferred labelling), which means these plans are justified and according to Definition 4.17, they are better than π_4: $(\pi_1, \pi_4), (\pi_2, \pi_4), (\pi_3, \pi_4) \in >_\pi$. Both plans π_1 and π_2 goal-dominate plan π_3, because they satisfy one goal more than π_3, so we have: $(\pi_1, \pi_3), (\pi_2, \pi_3) \in >_G$. Plan π_1 and π_2 satisfy the same set of goals, thus: $(\pi_1, \pi_2) \in \sim_G$. However, the number of norms violated in π_2 is greater than in π_1. Therefore, π_1 norm-dominates π_2: $(\pi_1, \pi_2) \in >_N$. Based on the third condition of Definition 4.17, $(\pi_1, \pi_2) \in >_\pi$. Since π_1 is justified and there is no plan better than π_1, $\nexists \pi$ s.t. $(\pi, \pi_1) \in \geq$, it is the best plan.

Consider the same plans again but now assume the agent prefers complying with $n3$ rather than satisfying $runningHospital$: $(n3, runningHospital) \geq^{gn}$. Here the effect of preferences on the justifiability of plans is remarkable: in addition to π_4, π_1 and π_2 are not justified and therefore, plan π_3 is the only justified plan and thus the best plan.

The above example shows how the agent can decide on the justifiability of the plans and use its preferences to resolve the conflict and identify the best course of action to follow.

6 RELATED WORK AND DISCUSSION

Several work consider norms and their conflict in planning and plan selection. [dos Santos et al. 2018] provides a comprehensive survey of normative conflict detection and resolution, including works that look at this problem in the context of planning which are reviewed below.

The BOID (Belief-Obligation-Intention-Desire) architecture [Broersen et al. 2002] extends the BDI architecture [Rao and Georgeff 1995] with the concept of obligation and uses agent types such as social, selfish, etc. to handle the conflicts between beliefs, desires, intentions and obligations. For instance if the agent is selfish, it will always considers its desires prior to any obligation. In contrast, a social agent always puts obligations prior to its desires. This architecture is considered as a model for norm-governed agent, although it lacks a computational model for implementation.

NoA, proposed by [Kollingbaum 2005], is a normative language and agent architecture. As a language, it specifies the normative concepts of obligation, prohibition and permission to regulate a specific type of agents interaction called "supervised interaction". As a practical reasoning agent architecture, it describes how agents select a plan from a pre-generated plan library such that the norms imposed on the agent at each point of time are obeyed. The agents do not have internal motivations such as goals or values that might conflict with norms, which therefore, enables the agent always to comply with norms.

[Belchior et al. 2018] propose a planning algorithm that in each step of planning checks whether the action added to the current plan would cause any normative conflict. If so, the action will be removed from the set of available actions to the agent. Thus, the plans generated are guaranteed to be conflict-free. That said, if all the actions available to the agent create normative conflict, the planning algorithm does not suggest any plans.

In order to generate conflict-free plans [Belchior et al. 2018; Broersen et al. 2002; Kollingbaum 2005] aim at resolving all conflicts. While generating conflict-free plans, due to the possibility of violation, we allow more freedom in action selection, where an agent can take an action that may cause a normative conflict. Alternative ways of resolving the conflict (violating one of the norms in conflict or another, or both) generate different plans that will then be subjected to comparison in terms of their goal satisfaction and normative quality. In addition, the focus of these approaches is on answering how an agent should act in a normative environment while it has conflicting goals and norms regardless of the transparency of the reasoning taken to answer that question. In contrast,we consider domains where humans

may need to understand why some action or plan was selected for execution. This requires a transparent reasoning mechanism, rather than the numerical utilities, that can serve as the basis for the justification of agent behaviour. We utilise formal argumentation to derive such a reasoning process. Thus the rest of this section is dedicated to surveying argumentation-based approaches to planning and practical reasoning.

Current work on argumentation-based practical reasoning can be divided into two categories, namely logic-based (e.g., [Amgoud 2003; Amgoud et al. 2008b; Hulstijn and van der Torre 2004; Rahwan and Amgoud 2006]) and scheme-based (e.g., [Atkinson and Bench-Capon 2007; Oren 2013]). In the former category, some argumentation semantics applied to an argumentation framework are used to generate a subset of consistent desires and plans to achieve them, which are in some sense optimised. The second category, into which this paper falls, utilizes defeasible inference rules (i.e., argument schemes and critical questions) to identify and justify some set of "best" plans. This latter category of approaches seeks to identify what plans can be constructed, ensuring consistency, correctness, and some form of optimality (e.g., maximal goal achievement), while specifying the order in which plan actions should be executed.

In the logic-based category, [Rahwan and Amgoud 2006] offers an instantiation of Dung's AF for generating consistent desires and plans for BDI agents [Rao and Georgeff 1995]. The authors consider three different Dung style AFs for arguing about beliefs and their truth value, about desires and justification of their adoption and about intentions. Arguing about intention, i.e., what is the best course of actions to achieve desires, is based on the utility of desires and resources required to achieve them. Continuing the work of [Rahwan and Amgoud 2006], [Amgoud et al. 2008b] proposes a constrained argumentation system that takes arguing about desires further by excluding the possibility of adopting desires that are not feasible. However, [Amgoud et al. 2008b] does not include any mechanism to compare various sets of justified and feasible desires. Unlike [Amgoud 2003; Rahwan and Amgoud 2006], [Hulstijn and van der Torre 2004] does not use multiple AFs to capture the conflicts between beliefs, desires/goals and intentions/plans. Instead, they extract goals by reasoning forward from desires, followed by deriving plans for goals, using planning rules. Goals that have a plan associated with them, can be modelled as an argument consisting of a claim and its necessary support. These arguments form an AF for planning, in which there is an attack between conflicting plans. They then look for an extension of this AF that maximises the number of achieved desires as opposed to considering the quality or utility of these desires that is the base of comparison in [Rahwan and Amgoud 2006].

Scheduling actions is not a focus of the logic-based approaches reviewed above. Instead, they concentrate on identifying a subset of consistent desires and the plans to achieve them. These approaches do not detail when and in which order agents should execute the selected plans. In contrast, recent developments in argumentation-based deliberative dialogues directly concern themselves with planning (e.g., [Belesiotis et al. 2010; Ferrando and Onaindia 2017]). However, they mainly focus on multi-agent plan construction and selection towards the achievement of a common goal when agents have different beliefs. Thus, dealing with conflicting goals does not figure in these approaches.

In the scheme-based category, most approaches to practical reasoning build on [Atkinson and Bench-Capon 2007], which uses Action-based Alternating Transition System (AATS) [Wooldridge and van der Hoek 2005], based on the agent's knowledge of actions with pre- and postconditions, and the values they promote. Using AATS along with a set of argument schemes and critical questions, arguments are generated for each available action. These arguments are then organised into a Value-based Argumentation Framework (VAF) [Bench-Capon 2003], where the preference between arguments is defined according to the values the actions promote and the goals they contribute to.

The approach proposed by [Oren 2013] is also based on AATS and an argumentation scheme, and adopts some ideas from [Atkinson and Bench-Capon 2007], however, unlike [Atkinson and Bench-Capon 2007], it permits practical reasoning in the presence of norms. As a result, preferences between arguments are defined by considering all possible

	Foundation	AF	Sociality	Goals	Conflict
[Rahwan and Amgoud 2006], [Amgoud et al. 2008b]	BDI	DAF	N/A	Achievement	Belief-Belief Desire-Desire Intention-Intention
[Hulstijn and van der Torre 2004]	BDI	DAF	N/A	Achievement	Goal-Goal, Plan-Plan
[Atkinson and Bench-Capon 2007], [Atkinson and Bench-Capon 2016], [Atkinson and Bench-Capon 2014]	AATS	VAF	Value	Achievement Maintenance	value-value
[Oren 2013]	AATS	ExAF[12]	Norm	Achievement Maintenance	Goal-Goal, Norm-Norm, Goal-Norm
[Toniolo et al. 2012]	SC[13]	BAF[14]	Norm	Achievement	Action-Action, Action-Plan, Action-Norm Goal-Goal
[Shams et al. 2016]	A STRIPS-based planning language	DAF	Norm	Achievement	Goal-Goal (static), Norm-Norm (temporal), Goal-Norm (static)
Current Work	A STRIPS-based planning language	DAF	Norm	Maintenance	Goal-Goal (temporal), Norm-Norm (temporal), Goal-Norm (temporal)

Table 1. Argumentation-based Frameworks for Practical Reasoning

interactions between norms and goals instead of values and goals [Atkinson and Bench-Capon 2007]. For an extensive discussion about the relation between norms and values, we refer the readers to [Bench-Capon 2016]. While [Oren 2013] provides a set of schemes for normative practical reasoning, the soundness and correctness of that approach was left for future work.

Similar to [Oren 2013], [Shams et al. 2016] constructs arguments for plans rather than actions. [Oren 2013] assumes that conflicts within and between goals and norms are inferred from sequences of states which come about due to action execution, and are thus left implicit, rather than being formally defined as is the case in [Shams et al. 2016] and here (see Section 3.3). Thus, although it is possible to explain why one path is preferred over another, it is not possible to explicitly link goal satisfaction with norm violation, unless all paths where the goal is satisfied are considered. In contrast, [Shams et al. 2016] explicitly considers why an agent does not satisfy a goal, or violate a norm, by appealing to the underlying conflict between them, rather than looking at all plans and inferring that, for example, if two goals are not simultaneously satisfied in any plan, then they are in conflict.

The work of [Toniolo et al. 2012] also considers norms in collaborative planning, but unlike our work and [Oren 2013], the norms are regimented, forcing the agent always to comply with norms, and ignoring the possibility of violation. Permitting violations of norms allows an agent to ignore a norm in order to pursue a more important goal, to deal with normative conflict, or to allow it to act in an unexpected situation. Furthermore, allowing violation is important in open multi-agent systems, where the unknown nature of agents participating in the system means that no guarantees regarding norm compliance can be provided.

Above we contrasted [Shams et al. 2016] with other scheme-based approaches [Atkinson and Bench-Capon 2007; Oren 2013; Toniolo et al. 2012]. Here the extensions that the current work is made to [Shams et al. 2016] are reviewed. First, this work considers maintenance rather than achievement goals [Hindriks and van Riemsdijk 2007] used in [Shams et al. 2016]. Since goals did not have temporal properties in [Shams et al. 2016], addressing goal-goal and goal-norm

[12] ExAF stands for Extended Argumentation Framework [Modgil 2007].
[13] SC stands for Situation Calculus [Reiter 1991].
[14] BAF stands for Bipolar Argumentation Framework [Amgoud et al. 2008a].

conflicts temporally was not an option. Thus, the second and more important extension that this work makes to [Shams et al. 2016] is addressing all types of conflict temporally, such that the plans generated are guaranteed to be free of goal-goal, norm-norm and goal-norm conflicts. An overall comparison of the current work to other related work is summarised in Table 1.

7 CONCLUSIONS

In this paper, we develop an approach which enables an agent to reason about its practical attitudes (i.e., actions, goals, norms) so as to identify the best course of actions to execute. Reasoning about what to do when the agent has multiple goals is a challenging task, particularly when these goals conflict. Moreover, social agents are often subject to norms imposed on them by the society of which they are members, or by other agents in the system. These norms aim to regulate the agent's behaviour in accordance with what is expected of them by others. However, these norms may not be aligned with the agent's goals, and can also be inconsistent. In such a complex environment the agent is not likely to be able to satisfy all its goals while complying with all the norms imposed on it. What is required of a rational agent is to be able to reason about what to do with respect to both its goals, and the norms imposed upon it, before committing to any course of action.

Argumentation serves as an effective computational tool for automated reasoning [Amgoud 2003; Bench-Capon et al. 2009; Dung 1995; Gaertner and Toni 2007; Oren et al. 2007]. In this role, argumentation is particularly important because it allows the reasoner to obtain consistent conclusions from conflicting, inconsistent and incomplete information. In the current work, we use formal argumentation techniques to reason about an agent's goals and norms, with the aim of identifying the best course of action - in terms of the quality (i.e., preferences) and quantity (i.e., number) of goals satisfied and norms violated - for the agent to follow.

7.1 Limitations

Several aspects of this work could be generalised further. Regarding norms, currently the activation condition of a norm is an action and the deactivation is a deadline. Alternatively, as done in other approaches (e.g., [Oren et al. 2008]), activation and deactivation of norms can occur when certain states are instantiated giving giving rise to state-based norms as opposed to the action-based ones here. Other work (e.g., [De Vos et al. 2013]) allow both state-based and action-based norms to co-exist. Given such state-based norms, additional propositions could be introduced, which hold when a norm is violated, activated or deactivated [Oren et al. 2008]. In such a situation, the status of a norm could then be used to trigger another norm, such as in the hierarchical normative framework described in [King et al. 2017], where a first-order norm can trigger a second order norm etc., enabling the capture of notions such as a prohibition to put certain obligations on actors in certain circumstances. We leave such extensions of our framework to future work. In addition to allowing alternatives for expressing activation/deactivation conditions in norm representation, if the framework proposed here is to be used in a multi-agent setting, the norm representation has to accommodate *roles* [Vázquez-Salceda et al. 2005] so that the agents know which norms are imposed on them. Since we are dealing with a single agent setting this is not an issue here.

Accommodating the above alternatives will have implications in terms of conflict detection and resolution, an extensive survey for which can be found in [dos Santos et al. 2018]. We briefly mention these implications: (i) Allowing state-based norms, from a technical viewpoint, is straightforward in our framework because it boils down to recognising a certain state — as is for example the case when dealing with goal achievement — which is associated with norm activation or deactivation. Detecting and resolving conflict, however, will then require defining mutually exclusive

states (perhaps via additional meta-level propositions). Once mutually exclusive states are defined, obligations that require the agent to achieve mutually exclusive states will be in direct conflict and so will be the obligations and prohibitions that oblige and prohibit from achieving the same state of affairs simultaneously. (ii) Allowing norms to act as activation/deactivation conditions of each other follows directly from the previous point, and can potentially be dealt with using the same machinery. In hierarchical frameworks such as the one put forward by [King et al. 2017], the assumption is that an higher order institution has power over a lower-order one, mimicking the roles of primary (higher), secondary (lower) legislation, and regulation/business processes (lower still). However, this does not mean higher level norms always override the lower level ones in the face of conflict; a lower level violation may uncover an exception or dispensation that was overlooked in the drafting of the higher level. (iii) Allowing roles can potentially give rise to direct and indirect conflicts that are caused by norms associated with roles that an agent plays [Cholvy and Cuppens 1995; Günay and Yolum 2013]. For example it is possible that one role of the agent obliges it to do an action, while a second role obliges the negation of the former action, in which case there will be a conflict between the two. Conflict caused by the norms associated with a single role are considered as direct, while those caused by norms associated with different roles are indirect.

We intend to address the mentioned types of conflict after accommodating the proposed generalisation to norm representation. A separate type of conflict that is possible to investigate even within the current representation, is the indirect one caused by side effects of plans. Termination of actions constantly changes the state the agent is in as a plan unfolds. Since actions can be in progress in parallel, a state might be influenced by multiple actions ending simultaneously. Thus, in addition to analysing the consequence of actions, consequence and side effects of plans can be analysed when detecting indirect normative conflict. [Kollingbaum 2005] successfully deals with detecting and resolving this type of conflict.

Another limitation is the assumption that the environment is deterministic (i.e.; the next state is predictable given knowledge of the previous state and the agent's action) and that actions are atomic and cannot be interrupted. In a dynamic environment and in particular in a multi-agent setting, an agent's actions may be interrupted and fail to complete. Using agent programming languages such as [Bordini et al. 2007] can facilitate this.

The final discussion point is about achievement goals. Here we are considering achievement goals that need to be satisfied at a certain point in time. For example $organiseSurvivorCamp = \left\langle \left\{ \begin{array}{c} areaSecured, \\ shelterBuilt \end{array} \right\}, 15, 15 \right\rangle$ has to be satisfied at time 15. This could have been made more flexible if achievement goals meant to be satisfied at some point *before* the deadline. Considering achievement goals that have to be satisfied before a certain point in time makes it very difficult to pinpoint goal-goal and goal-norm conflict, as in these cases we must consider a time frame during which the requirements of an achievement goal should hold. The fact that a goal was not satisfied at any time point before the deadline can have many reasons that differ at different times (i.e., the reason why the goal was not satisfied at time x might well be different from why it was not satisfied at time y). This ultimately makes it potentially impossible to find a specific justification for why an achievement goal was not satisfied before its deadline.

7.2 Future Work

In addition to reasoning, argumentation can also serve as an effective computational tool for generating explanation [Baroni and Giacomin 2009; Caminada et al. 2014a; Fan and Toni 2015; García et al. 2013; Lacave and Díez 2004; Shams et al. 2016]. Agents equipped with argumentation capabilities can explain the validity of their inferences or reasoning to humans, in the form of explanatory dialogues [Moulin et al. 2002]. These dialogues provide a dialectic proof mechanism

for argumentation semantics [Dung 1995] through the exchange of utterances between parties. In the future, we intend to give explanation for why a certain course of action is the best for an agent to follow, in particular focusing on temporal properties of goals and norms and how different sequencing of actions can cause or resolve certain conflicts with respect to time.

Another avenue that we envision for future work is to consider possible conflicts as well as the definite ones we have covered in this paper. For instance, we have defined conflict between an obligation and a goal for the extreme cases, where no scheduling of actions allows compliance with the obligation as well as satisfying the goal. However, some schedules could exist which make the possibility of compliance with the obligation and satisfying the goal more probable. These aspects can potentially be investigated within a Timed Argumentation Framework (TAF) [Budán et al. 2015] or Probabilistic Argumentation Framework [Li et al. 2011].

Appendix A VISUALISATION OF PLANS IN EXAMPLE 5.1

Visualisation of plans provided here, is in fact the visualisation of the answer set associated to that plan, that shows the actions executed in each state as well as the states that execution of actions brings about. The boxes under each state contain the fluents hold in that state. When a fluent holds for the first time, it appears in bold, but when it is carried forward from a previous state, it is not bold anymore. If a fluent is terminated in a state it will be crossed out and will not appear in the following state. To increase the readability norm fluents appear in a specific format, for instance, norm fluent $f(n_1, buildShelter, 5)$ states that execution of action *buildShelter* is forbidden till state 5. *cmp* fluents encode norm compliance and are highlighted in yellow, whereas violation fluents are encoded by *vol* and are highlighted in red. Finally, *sat* fluents (highlighted by green) encode a goal is satisfied in a specific state, however in order for a goal to be satisfied in a plan it has to be kept satisfied in all states that are included in the maintenance period.

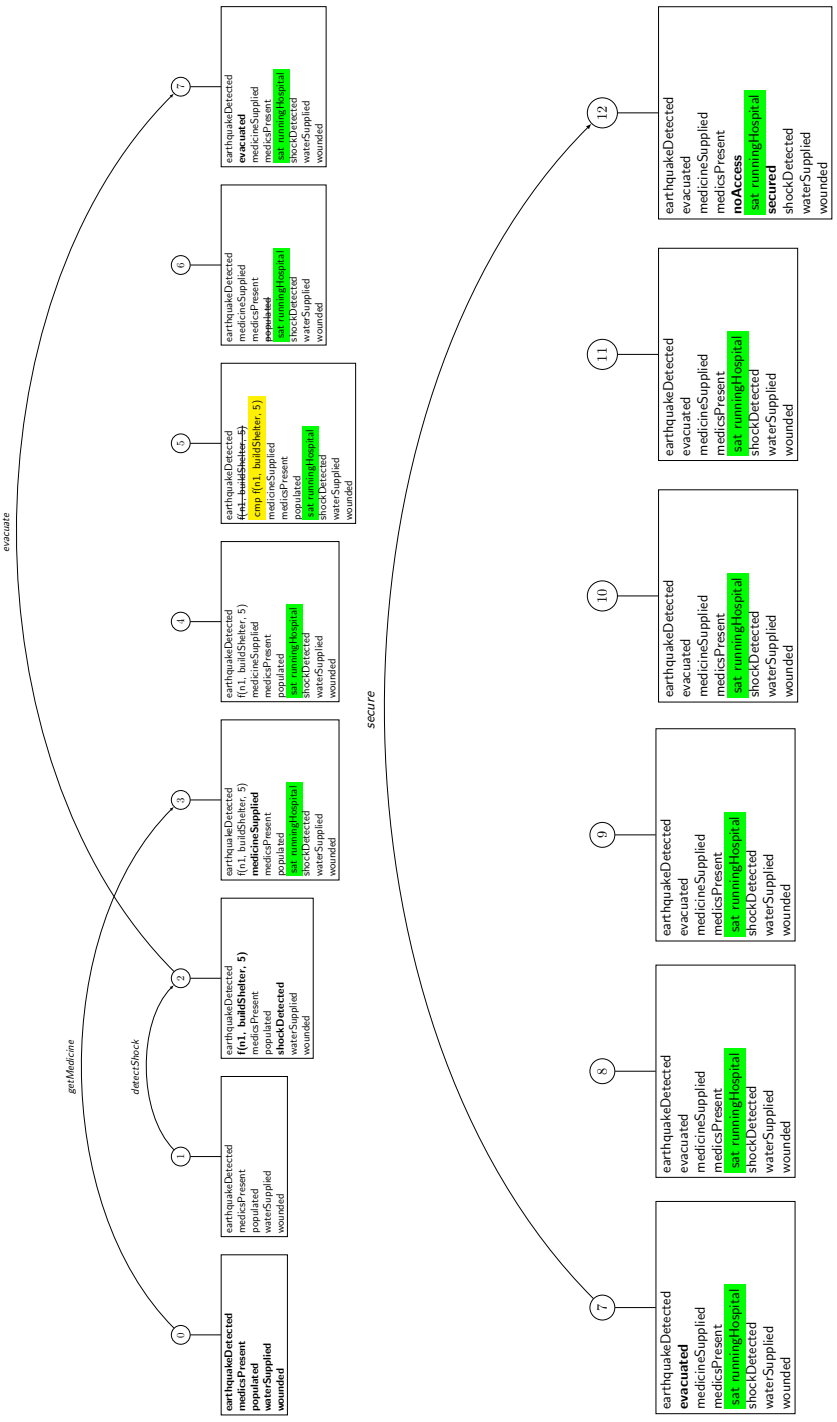

Fig. 5. Visualisation of Plan 1, states 0 to 12

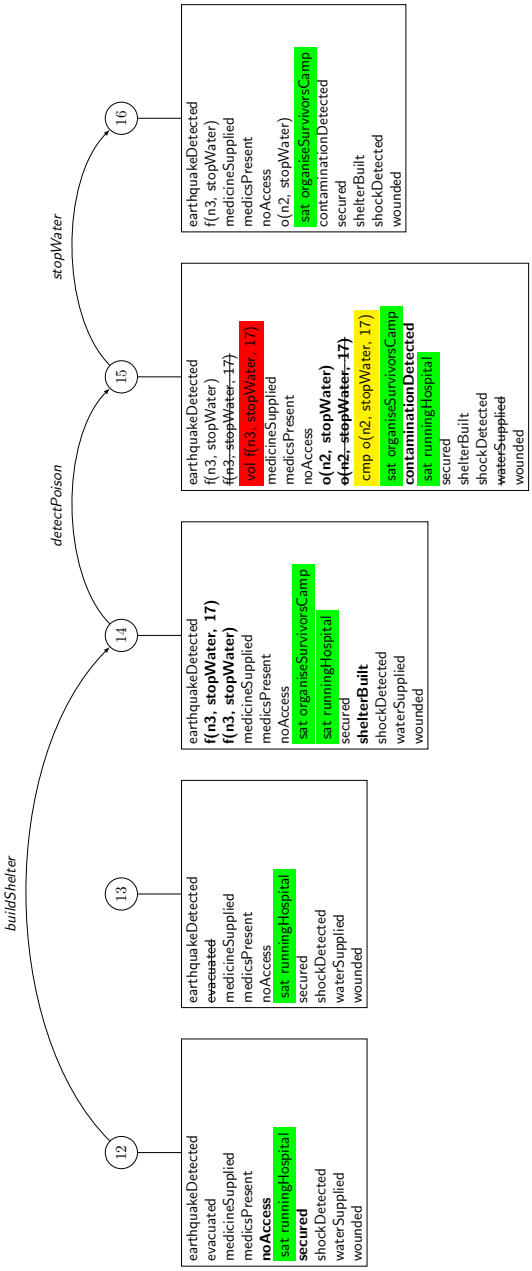

Fig. 6. Visualisation of Plan 1, states 12 to 16

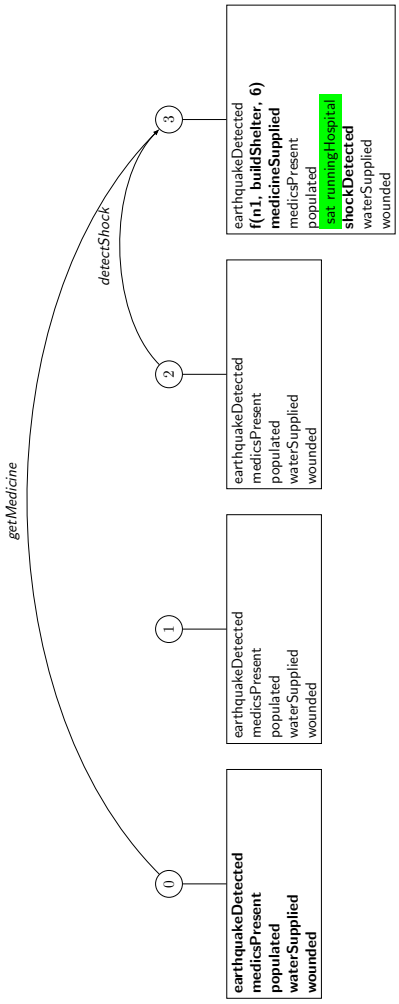

Fig. 7. Visualisation of Plan 2, states 0 to 3

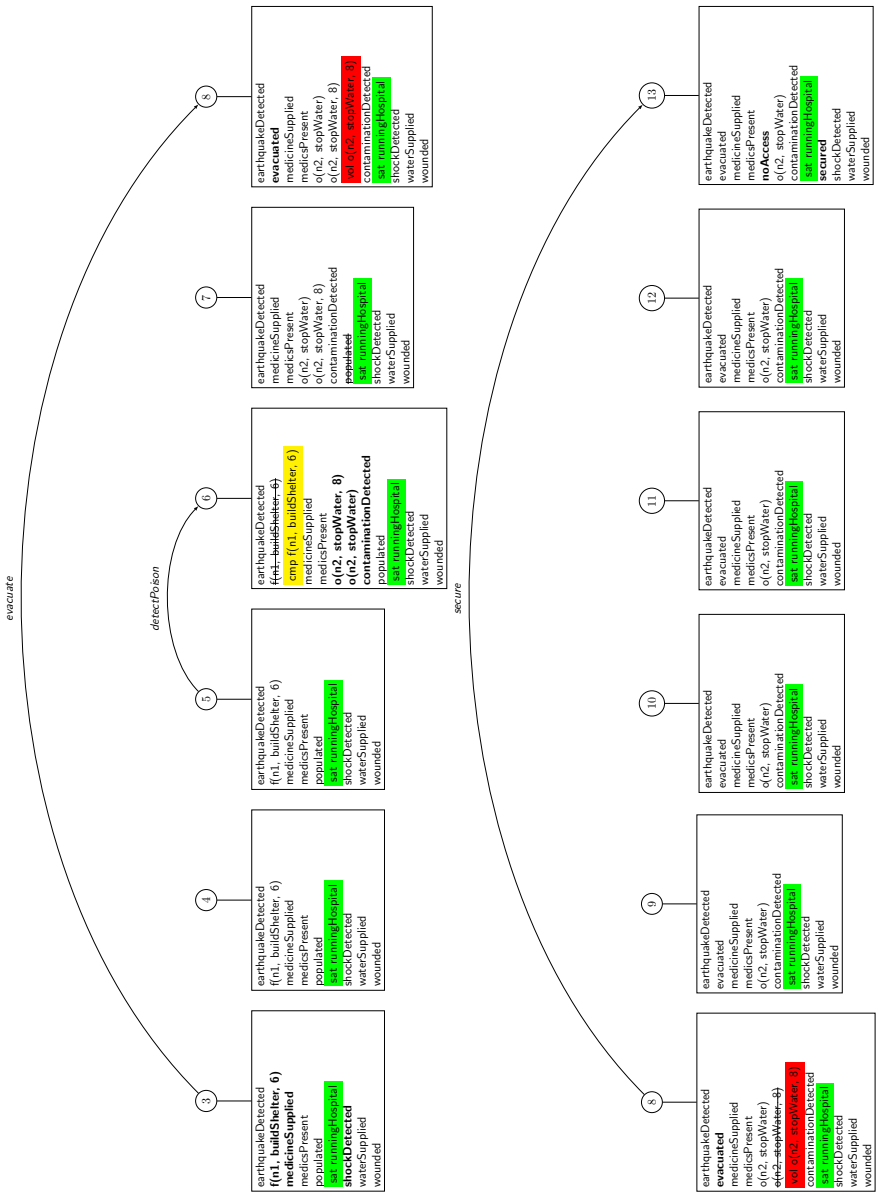

Fig. 8. Visualisation of Plan 2, states 3 to 13

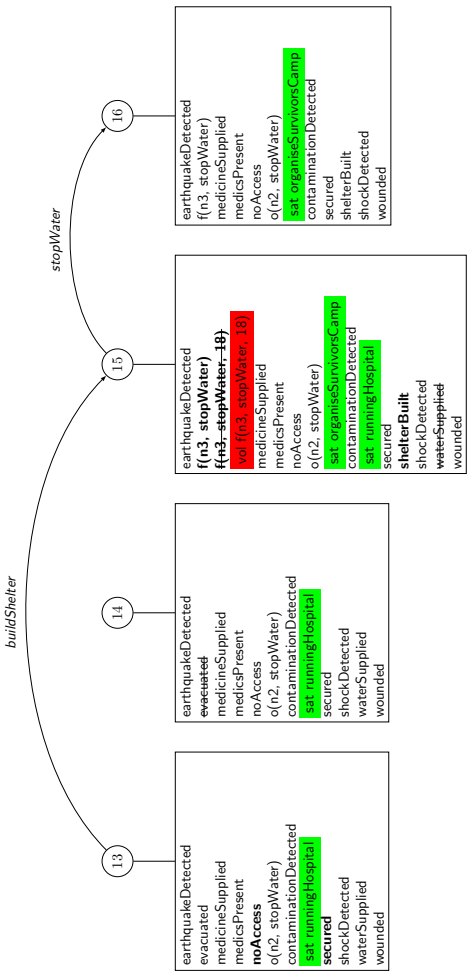

Fig. 9. Visualisation of Plan 2, states 13 to 16

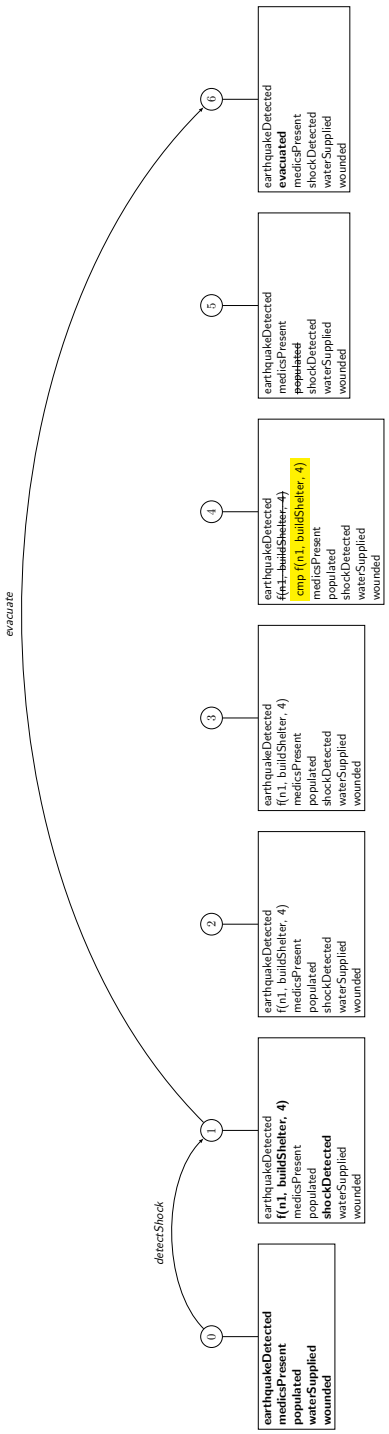

Fig. 10. Visualisation of Plan 3, states 0 to 6

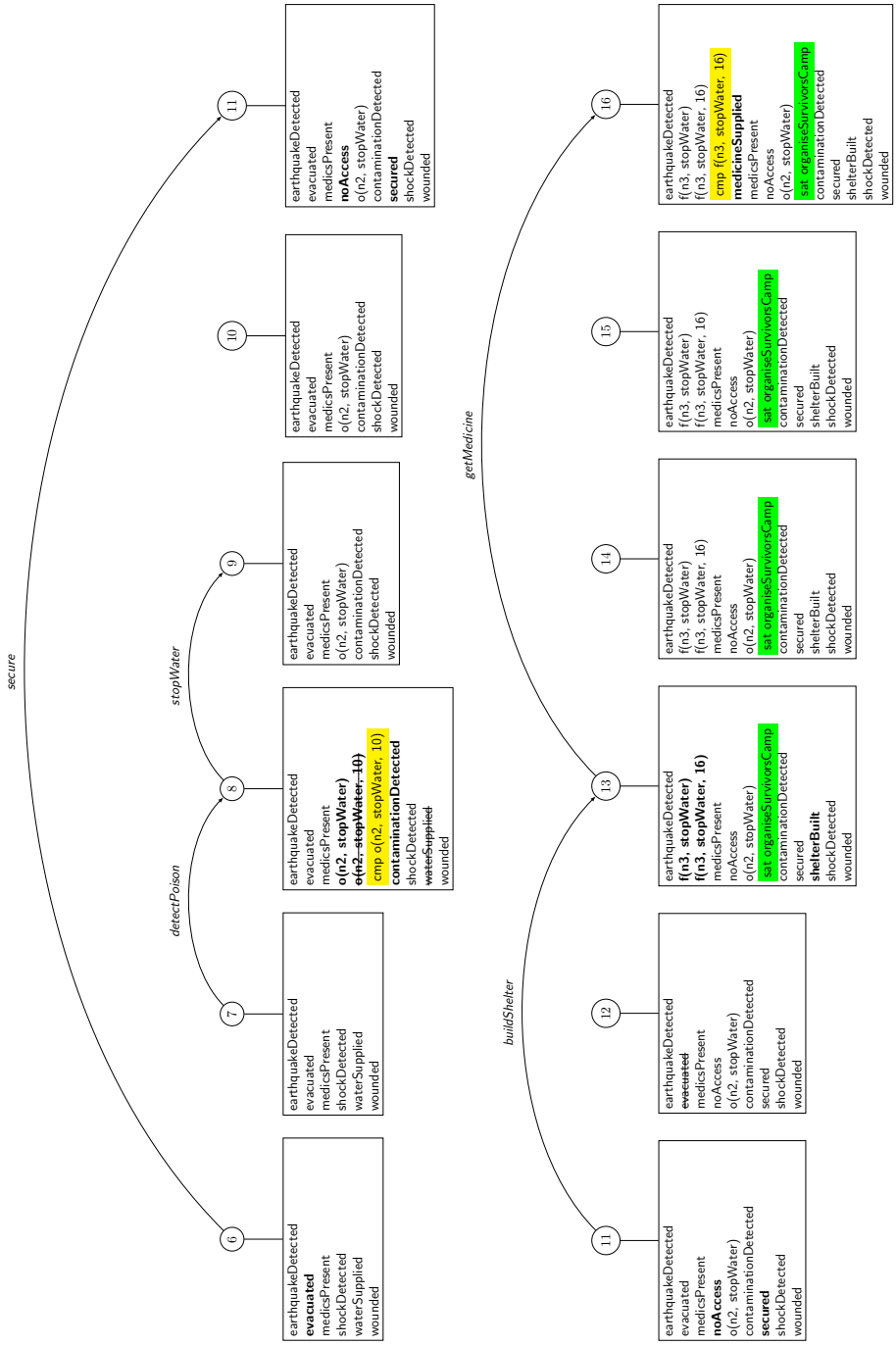

Fig. 11. Visualisation of Plan 3, states 6 to 16

Fig. 12. Visualisation of Plan 4, states 0 to 11

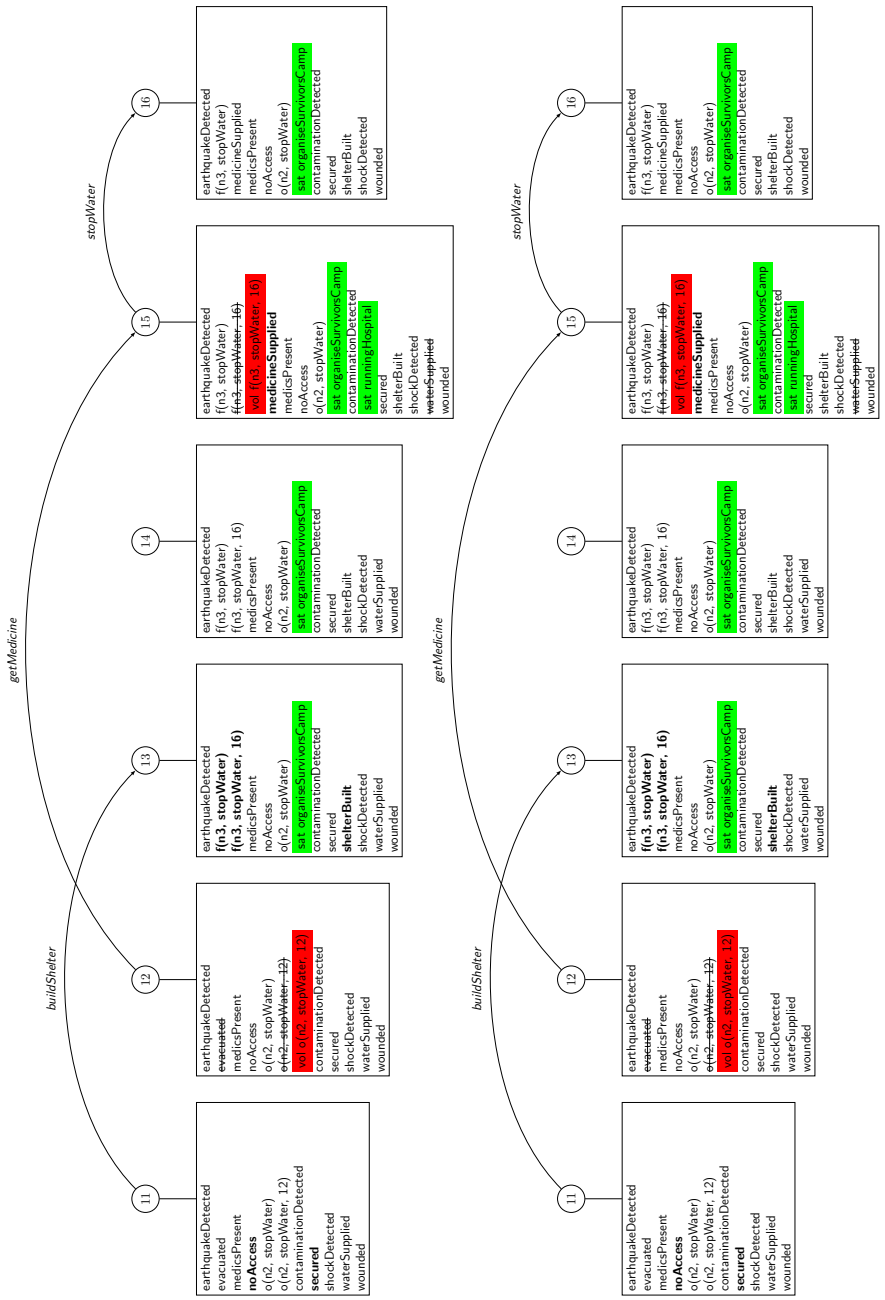

Fig. 13. Visualisation of Plan 4, states 6 to 16

Appendix B PROOFS

Proof of Theorem 4.11. Assume that $\trianglerighteq_G$ is a total preorder on $P(G)$, while $\succ^g$ is not a total order on G. The latter means that $\exists g, g' \in G$ s.t.$(g, g') \notin \succ^g$ and $(g', g) \notin \succ^g$. Since $\{g\}$ and $\{g'\}$ both belong to $P(G)$ and $(g, g') \notin \succ^g$ and $(g', g) \notin \succ^g$, we conclude that $\trianglerighteq_G$ is not a total preorder which is contrary to assumption.

Assume that $\succ^g$ is a total order on G. In oder to prove that $\trianglerighteq_G$ is a total preorder, we need to prove that it is reflexive, transitive, and total. It is easy to see it is reflexive and transitive (see Definition 4.9). Below we prove that it is total meaning that $\forall G_i, G_j \in P(G)$ either $(G_i, G_j) \in \trianglerighteq_G$ or $(G_j, G_i) \in \trianglerighteq_G$. Let $g = max((G_i \cup G_j) \setminus (G_i \cap G_j))$, where max denotes the most preferred element of a set with respect with a certain preference ordering:

case 1: $g \in G_i$. Then $\forall g' \in G_j, \exists g \in G_i$ s.t. $(g, g') \in \succ^g$, which means $(G_i, G_j) \in \trianglerighteq_G$.

case 2: $g \in G_j$. Then $\forall g' \in G_i, \exists g \in G_j$ s.t. $(g, g') \in \succ^g$, which means $(G_j, G_i) \in \trianglerighteq_G$.

□

Proof of Proposition 4.18. We need to show: $\forall \pi \in \Pi, (\pi, \pi) \notin >_\pi$.

π cannot be justified and unjustified at the same time. Also, both $>_G$ and $>_N$ are irreflexive so $(\pi, \pi) \notin >_G$, and $(\pi, \pi) \notin >_N$. Thus, $(\pi, \pi) \notin >_\pi$. □

Proof of Proposition 4.19. We need to show: $\forall \pi_1, \pi_2 \in \Pi$, if $\pi_1 \neq \pi_2$ and $(\pi_1, \pi_2) \in >_\pi$ then $(\pi_2, \pi_1) \notin >_\pi$.
Assume that $\pi_1 \neq \pi_2$ and $(\pi_1, \pi_2) \in >_\pi$ while $(\pi_2, \pi_1) \in >_\pi$.

case 1: $(\pi_2, \pi_1) \in >_\pi$ because π_2 is justified and π_1 is not. This means that $(\pi_1, \pi_2) \notin >_\pi$ which is contrary to assumption.

case 2: $(\pi_2, \pi_1) \in >_\pi$ because π_2 and π_1 are both justified and $(\pi_2, \pi_1) \in >_G$. Since $>_G$ is antisymmetric $(\pi_1, \pi_2) \notin >_G$, which means that $(\pi_1, \pi_2) \notin >_\pi$ that is contrary to assumption.

case 3: $(\pi_2, \pi_1) \in >_\pi$ because π_2 and π_1 are both justified and $(\pi_2, \pi_1) \in \sim_G$ but $(\pi_2, \pi_1) \in >_N$. Since $\sim_G$ is symmetric, $(\pi_1, \pi_2) \in \sim_G$ and since $>_N$ is antisymmetric $(\pi_1, \pi_2) \notin >_N$, which means that $(\pi_1, \pi_2) \notin >_\pi$ that is contrary to assumption.

□

Proof of Proposition 4.20. We need to show: $\forall \pi_1, \pi_2, \pi_3 \in \Pi$, if $(\pi_1, \pi_2) \in >_\pi$ and $(\pi_2, \pi_3) \in >_\pi$, then $(\pi_1, \pi_3) \in >_\pi$.

case 1: $(\pi_1, \pi_2) \in >_\pi$ because π_1 is justified but π_2 is not. On the other hand, $(\pi_2, \pi_3) \in >_\pi$ because π_2 is justified but π_3 is not. This cannot be the case since π_2 cannot be both justified and unjustified.

case 2: $(\pi_1, \pi_2) \in >_\pi$ because π_1 is justified but π_2 is not. On the other hand, $(\pi_2, \pi_3) \in >_\pi$ because π_2 and π_3 are both justified but $(\pi_2, \pi_3) \in >_G$. This cannot be the case since π_2 cannot be both justified and unjustified.

case 3: $(\pi_1, \pi_2) \in >_\pi$ because π_1 is justified but π_2 is not. On the other hand, $(\pi_2, \pi_3) \in >_\pi$ because π_2 and π_3 are both justified and $(\pi_2, \pi_3) \in \sim_G$, while $(\pi_2, \pi_3) \in >_N$. This cannot be the case since π_2 cannot be both justified and unjustified.

case 4: $(\pi_1, \pi_2) \in >_\pi$ because π_1 and π_2 are both justified but $(\pi_1, \pi_2) \in >_G$. On the other hand, $(\pi_2, \pi_3) \in >_\pi$ because π_2 is justified but π_3 is not. Since π_1 is justified and π_3 is not, $(\pi_1, \pi_3) \in >_\pi$.

case 5: $(\pi_1, \pi_2) \in >_\pi$ because π_1 and π_2 are both justified but $(\pi_1, \pi_2) \in >_G$. On the other hand, $(\pi_2, \pi_3) \in >_\pi$ because π_2 and π_3 are both justified but $(\pi_2, \pi_3) \in >_G$. Since $>_G$ is transitive, from $(\pi_1, \pi_2) \in >_G$ and $(\pi_2, \pi_3) \in >_G$ we conclude that $(\pi_1, \pi_3) \in >_G$. Thus, $(\pi_1, \pi_3) \in >_\pi$.

case 6: $(\pi_1, \pi_2) \in >_\pi$ because π_1 and π_2 are both justified but $(\pi_1, \pi_2) \in >_G$. On the other hand, $(\pi_2, \pi_3) \in >_\pi$ because π_2 and π_3 are both justified and $(\pi_2, \pi_3) \in \sim_G$, while $(\pi_2, \pi_3) \in >_N$. Since $>_G$ and $\sim_G$ are both transitive, from $(\pi_1, \pi_2) \in >_G$ and $(\pi_2, \pi_3) \in \sim_G$ we conclude that $(\pi_1, \pi_3) \in >_G$. Thus, $(\pi_1, \pi_3) \in >_\pi$.

case 7: $(\pi_1, \pi_2) \in >_\pi$ because π_1 and π_2 are both justified and $(\pi_1, \pi_2) \in \sim_G$ but $(\pi_1, \pi_2) \in >_N$. On the other hand, $(\pi_2, \pi_3) \in >_\pi$ because π_2 is justified but π_3 is not. Since π_1 is justified and π_3 is not, $(\pi_1, \pi_3) \in >_\pi$.

case 8: $(\pi_1, \pi_2) \in >_\pi$ because π_1 and π_2 are both justified and $(\pi_1, \pi_2) \in \sim_G$ but $(\pi_1, \pi_2) \in >_N$. On the other hand, $(\pi_2, \pi_3) \in >_\pi$ because π_2 and π_3 are both justified but $(\pi_2, \pi_3) \in >_G$. Since $>_G$ and $\sim_G$ are both transitive, from $(\pi_1, \pi_2) \in \sim_G$ and $(\pi_2, \pi_3) \in >_G$ we conclude that $(\pi_1, \pi_3) \in >_G$. Thus, $(\pi_1, \pi_3) \in >_\pi$.

case 9: $(\pi_1, \pi_2) \in >_\pi$ because π_1 and π_2 are both justified and $(\pi_1, \pi_2) \in \sim_G$ but $(\pi_1, \pi_2) \in >_N$. On the other hand, $(\pi_2, \pi_3) \in >_\pi$ because π_2 and π_3 are both justified and and $(\pi_2, \pi_3) \in \sim_G$ and but $(\pi_2, \pi_3) \in >_N$. Since $\sim_G$ is transitive, from $(\pi_1, \pi_2) \in \sim_G$ and $(\pi_2, \pi_3) \in \sim_G$ we conclude that $(\pi_1, \pi_3) \in \sim_G$. Also since Since $>_N$ is transitive, from $(\pi_1, \pi_2) \in >_N$ and $(\pi_2, \pi_3) \in >_N$ we conclude that $(\pi_1, \pi_3) \in >_N$. Thus, $(\pi_1, \pi_3) \in >_\pi$.

□

Proof of Proposition 4.21. We need to show that $\sim_\pi$ is reflexive, symmetric and transitive.

Reflexive: Assume that $\exists \pi \in \Pi$ s.t. $(\pi, \pi) \notin \sim_\pi$. If $(\pi, \pi) \notin \sim_\pi$ then $(\pi, \pi) \in >_\pi$, which cannot be the case since $>_\pi$ is not reflexive.

Symmetric: Assume that $(\pi_1, \pi_2) \in \sim_\pi$ but $(\pi_2, \pi_1) \notin \sim_\pi$. If $(\pi_2, \pi_1) \notin \sim_\pi$ then either

case 1: $(\pi_1, \pi_2) \in >_\pi$, which means that $(\pi_1, \pi_2) \notin \sim_\pi$. This is contrary to assumption $(\pi_1, \pi_2) \in \sim_\pi$. Therefore, $(\pi_2, \pi_1) \in \sim_\pi$.

case 2: $(\pi_2, \pi_1) \in >_\pi$, which means that $(\pi_1, \pi_2) \notin \sim_\pi$. This is contrary to assumption $(\pi_1, \pi_2) \in \sim_\pi$. Therefore, $(\pi_2, \pi_1) \in \sim_\pi$.

Transitive: We need to show that if $(\pi_1, \pi_2) \in \sim_\pi$ and $(\pi_2, \pi_3) \in \sim_\pi$, then $(\pi_1, \pi_3) \in \sim_\pi$.

case 1: $(\pi_1, \pi_2) \in \sim_\pi$ because they are both not justified. Also $(\pi_2, \pi_3) \in \sim_\pi$ because they are both not justified. Therefore, neither of π_1 and π_3 are justified, so $(\pi_1, \pi_3) \in \sim_\pi$.

case 2: $(\pi_1, \pi_2) \in \sim_\pi$ because they are both not justified. On the other hand, $(\pi_2, \pi_3) \in \sim_\pi$ because they are both justified and $(\pi_2, \pi_3) \in \sim_G$ and $(\pi_2, \pi_3) \in \sim_N$. But π_2 cannot be both justified and unjustified.

case 3: $(\pi_1, \pi_2) \in \sim_\pi$ because they are both justified and $(\pi_1, \pi_2) \in \sim_G$ and $(\pi_1, \pi_2) \in \sim_N$. On the other hand, $(\pi_2, \pi_3) \in \sim_\pi$ because neither of them is justified. But π_2 cannot be both justified and unjustified.

case 4: $(\pi_1, \pi_2) \in \sim_\pi$ because they are both justified and $(\pi_1, \pi_2) \in \sim_G$ and $(\pi_1, \pi_2) \in \sim_N$. On the other hand, $(\pi_2, \pi_3) \in \sim_\pi$ because they are both justified and $(\pi_2, \pi_3) \in \sim_G$ and $(\pi_2, \pi_3) \in \sim_N$. Since $\sim_G$ and $\sim_N$ are both transitive, it follows that π_1 and π_3 are both justified and $(\pi_1, \pi_3) \in \sim_G$ and $(\pi_1, \pi_3) \in \sim_N$. Thus, $(\pi_1, \pi_3) \in \sim_\pi$.

□

Proof of Proposition 4.23. $\geq$ is a total order on Π iff it is antisymmetric, transitive and total.

- antisymmetric: We need to prove that if $([\pi_i], [\pi_j]) \in \geq$ and $([\pi_j], [\pi_i]) \in \geq$ then $[\pi_i] = [\pi_j]$. If $([\pi_i], [\pi_j]) \in \geq$, then $(\pi_i, \pi_j) \in >_\pi$ or $(\pi_i, \pi_j) \in \sim_\pi$. If $(\pi_i, \pi_j) \in >_\pi$, then $(\pi_j, \pi_i) \notin >_\pi$. Because $([\pi_j], [\pi_i]) \in \geq$, we have to have $(\pi_j, \pi_i) \in \sim_\pi$, which leads to $[\pi_j] = [\pi_i]$.
- transitive: We need to show that if $([\pi_i], [\pi_j]) \in \geq$ and $([\pi_j], [\pi_i]) \in \geq$, then $([\pi_i], [\pi_k]) \in \geq$. If $([\pi_i], [\pi_j]) \in \geq$ then $(\pi_i, \pi_j) \in >_\pi$ or $(\pi_i, \pi_j) \in \sim_\pi$. Similarly, if $([\pi_j], [\pi_k]) \in \geq$ then $(\pi_j, \pi_k) \in >_\pi$ or $(\pi_j, \pi_k) \in \sim_\pi$. Since $>_\pi$ are $\sim_\pi$ are both transitive, in either of the following four cases we conclude that $([\pi_i], [\pi_k]) \geq$:

case 1: $(\pi_i, \pi_j) \in >_\pi$, and $(\pi_j, \pi_k) \in >_\pi$ implies $(\pi_i, \pi_k) \in >_\pi$ and therefore $([\pi_i], [\pi_k)] \in \geq$.
case 2: $(\pi_i, \pi_j) \in >_\pi$, and $(\pi_j, \pi_k) \in \sim_\pi$ implies $(\pi_i, \pi_k) \in >_\pi$ and therefore $([\pi_i], [\pi_k)] \in \geq$.
case 3: $(\pi_i, \pi_j) \in \sim_\pi$, and $(\pi_j, \pi_k) \in >_\pi$ implies $(\pi_i, \pi_k) \in >_\pi$ and therefore $([\pi_i], [\pi_k)] \in \geq$.
case 4: $(\pi_i, \pi_j) \in \sim_\pi$, and $(\pi_j, \pi_k) \in \sim_\pi$ implies $(\pi_i, \pi_k) \in \sim_\pi$ and therefore $([\pi_i], [\pi_k)] \in \geq$.

- total: If $\geq$ is not total, then $\exists [\pi_i], [\pi_j]$ s.t. $([\pi_i], [\pi_j]) \notin \geq$ and $([\pi_j], [\pi_i]) \notin \geq$. If $([\pi_i], [\pi_j]) \notin \geq$, then $(\pi_i, \pi_j) \notin >_\pi$ and $(\pi_i, \pi_j) \notin \sim_\pi$. On the other hand, if $([\pi_j], [\pi_i]) \notin \geq$, then $(\pi_j, \pi_i) \notin >_\pi$ and $(\pi_j, \pi_i) \notin \sim_\pi$. From $(\pi_i, \pi_j) \notin >_\pi$ and $(\pi_j, \pi_i) \notin >_\pi$ we conclude that $(\pi_i, \pi_j) \in \sim_\pi$, which is contradictory to $(\pi_i, \pi_j) \notin \sim_\pi$ and $(\pi_j, \pi_i) \notin \sim_\pi$.

□

Proof of Property 1. Since all arguments are built on defeasible rules, the property follows immediately. □

Proof of Property 2. Suppose the conclusions of extension E are inconsistent, i.e., there are arguments $Arg_\alpha, Arg_\beta \in E$ such that:
- Arg_α's conclusion requires executing plan π and Arg_β's conclusion requires satisfying goal g/complying with norm n, while g is not satisfied/n is violated in π. Thus, Arg_β defeats Arg_α; E is not conflict-free and cannot be an extension.
- Arg_α's conclusion requires satisfying goal g/complying with norm n and Arg_β's conclusion requires satisfying goal $g$fi/complying with norm n', while g/n and g'/n' are inconsistent. Thus, Arg_α attacks Arg_β and vice versa. Due to the preferences, at least one of these attacks is identified as defeat and therefore E is not conflict-free and not an extension. □

Proof of Property 3. Follows immediately from lack of strict rules. □

Proof of Property 4. Every preferred labelling is a complete labelling. An argument is labelled *in* by a complete labelling iff all its attackers are labelled *out*. Therefore, a plan argument is labelled *in* by a preferred labelling iff all its attackers, namely the arguments for goals that it does not satisfy and norms that it violates, are labelled *out* by that labelling. □

Proof of Property 5. Since $Arg_\pi \in in(\mathcal{L})$, from Property 4 we know that $\bigcup_{g \in G \setminus G_\pi} Arg_g \cup \bigcup_{n \in N_{vol(\pi)}} Arg_n \subseteq out(\mathcal{L})$. We also know from the definition of a plan that $\bigcup_{g \in G_\pi} Arg_g \cup \bigcup_{n \in N_{cmp(\pi)}} Arg_n$ is conflict free. Since all possible attackers of $\bigcup_{g \in G_\pi} Arg_g \cup \bigcup_{n \in N_{cmp(\pi)}} Arg_n$ belong to $\bigcup_{g \in G \setminus G_\pi} Arg_g \cup \bigcup_{n \in N_{vol(\pi)}} Arg_n$ and $\bigcup_{g \in G \setminus G_\pi} Arg_g \cup \bigcup_{n \in N_{vol(\pi)}} Arg_n$ are all labelled *out*, we conclude that $\bigcup_{g \in G_\pi} Arg_g \cup \bigcup_{n \in N_{cmp(\pi)}} Arg_n \subseteq in(\mathcal{L})$. □

Proof of Property 6. From Property 4 and Property 5 we know that if $Arg_\pi \in in(\mathcal{L})$ then $\bigcup_{g \in G \setminus G_\pi} Arg_g \cup \bigcup_{n \in N_{vol(\pi)}} Arg_n \subseteq out(\mathcal{L})$ and $\bigcup_{g \in G_\pi} Arg_g \cup \bigcup_{n \in N_{cmp(\pi)}} Arg_n \subseteq in(\mathcal{L})$. Since every preferred labelling is a complete labelling and the following property holds for complete labellings: if $out(\mathcal{L}_{cmp1}) = out(\mathcal{L}_{cmp2})$ then $\mathcal{L}_{cmp1} = \mathcal{L}_{cmp2}$; we conclude that there is no more than one preferred labelling in which $Arg_\pi \in in(\mathcal{L})$. □

Proof of Property 7. In Property 4 we showed that if $Arg_\pi \in in(\mathcal{L})$ then $\bigcup_{g \in G \setminus G_\pi} Arg_g \cup \bigcup_{n \in N_{vol(\pi)}} Arg_n \subseteq out(\mathcal{L})$ and $\bigcup_{g \in G_\pi} Arg_g \cup \bigcup_{n \in N_{cmp(\pi)}} Arg_n \subseteq in(\mathcal{L})$, which makes the $undec(\mathcal{L}) = \emptyset$. A preferred labelling with $undec(\mathcal{L}) = \emptyset$ is a stable labelling. Therefore, $\mathcal{L}$ is a stable labelling. □

Proof of Property 8. Elements of set $Pref(Arg)$ cannot be defeated, as the set is conflict-free and the remaining arguments belong to $Arg \setminus Pref(Arg)$. The latter cannot defeat elements of $Pref(Arg)$, because this would imply an attack from a less preferred argument to a more preferred one has resulted in a defeat, which is contrary to

assumption. Assume that $\exists Arg_\alpha \in Pref(Arg)$ such that $Arg_\alpha \notin in(\mathcal{L})$. If $\nexists Arg_\beta \in in(\mathcal{L})$ s.t. $(Arg_\alpha, Arg_\beta) \in Def$ then Arg_α should have been labelled *in* by $\mathcal{L}$ otherwise it is contrary to the assumption of maximality of preferred labellings. If $\exists Arg_\beta \in in(\mathcal{L})$ s.t. $(Arg_\alpha, Arg_\beta) \in Def$ then $\exists Arg_\gamma \in in(\mathcal{L})$ s.t. $(Arg_\gamma, Arg_\alpha) \in Def$, which is contradictory to the fact that Arg_α cannot be defeated. Therefore, all elements of $Pref(Arg)$ are labelled in by $in(\mathcal{L})$. □

Proof of Property 9. plan argument aside, E_1 and E_2 differ in one argument and that is $Arg_\alpha \in E_1$ and $Arg_\beta \in E_2$. Arg_α and Arg_β are either goal arguments or norm arguments. So we have the following cases:

(1) $\alpha, \beta \in G$: Since Arg_α and Arg_β are both goal arguments and $(Arg_\alpha, Arg_\beta) \in >$, we conclude that plan π_1 goal-dominates plan π_2 and therefore, $(\pi_1, \pi_2) \in >_\pi$.

(2) $\alpha, \beta \in N$: Since Arg_α and Arg_β are both norm arguments, and $(Arg_\alpha, Arg_\beta) \in >$, plan π_1 norm-dominates plan π_2 and since none of them goal-dominates the other one (we know that because the set of goal arguments is the same in E_1 and E_2), we therefore have: $(\pi_1, \pi_2) \in >_\pi$.

(3) $\alpha \in G$ and $\beta \in N$: Since Arg_α is a goal argument and Arg_β is a norm argument, we conclude that plan π_1 satisfies a goal in addition to the goals satisfied in plan π_2, therefore, plan π_1 goal-dominates plan π_2 and we have $(\pi_1, \pi_2) \in >_\pi$.

(4) $\alpha \in N$ and $\beta \in G$: Similar to previous case, here we conclude that plan π_2 satisfies a goal in addition to the goals satisfied in plan π_1, therefore, plan π_2 goal-dominates plan π_1 and we have $(\pi_2, \pi_1) \in >_\pi$. The result is not intuitive, since the extensions differ in one argument and the argument preference suggests that π_1 should be better than π_2, however we have the opposite. The reason that this property does not hold in this case, is the fact that when ranking plans (Definition 4.17) goal-dominance take precedence over norm-dominance and therefore, the following preference $(Arg_\alpha, Arg_\beta) \in >$ is overridden.

□

REFERENCES

Alan S. Abrahams and Jean M. Bacon. 2002. The Life and Times of Identified, Situated, and Conflicting Norms. In *International Workshop on Deontic Logic in Computer Science*. 3–20.

Leila Amgoud. 2003. A Formal Framework for Handling Conflicting Desires. In *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (LNCS)*, Vol. 2711. Springer, 552–563.

Leila Amgoud and Claudette Cayrol. 2002. A Reasoning Model Based on the Production of Acceptable Arguments. *Annals of Mathematics Artificial Intelligence* 34, 1-3 (2002), 197–215.

Leila Amgoud, Claudette Cayrol, Marie-Christine Lagasquie-Schiex, and P. Livet. 2008a. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems* 23, 10 (2008), 1062–1093.

Leila Amgoud, Caroline Devred, and Marie-Christine Lagasquie-Schiex. 2008b. A Constrained Argumentation System for Practical Reasoning. In *International Workshop on Argumentation in Multi-Agent Systems (LNCS)*, Vol. 5384. Springer, 37–56.

Leila Amgoud and Henri Prade. 2009. Using arguments for making and explaining decisions. *Artificial Intelligence* 173, 3-4 (2009), 413–436.

Leila Amgoud and Srdjan Vesic. 2014. Rich preference-based argumentation frameworks. *International Journal of Approximate Reasoning* 55, 2 (2014), 585–606.

Mukta S. Aphale, Timothy J. Norman, and Murat Sensoy. 2014. Goal directed policy conflict detection and prioritisation: an empirical evaluation. In *International conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2014)*. IFAAMAS/ACM, 1489–1490.

Katie Atkinson. 2005. *What should we do?: Computational representation of persuasive argument in practical reasoning*. Ph.D. Dissertation.

Katie Atkinson and Trevor Bench-Capon. 2007. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence* 171, 10-15 (2007), 855–874.

Katie Atkinson and Trevor Bench-Capon. 2014. Taking the Long View: Looking Ahead in Practical Reasoning. In *Computational Models of Argument (Frontiers in Artificial Intelligence and Applications)*, Vol. 266. IOS Press, 109–120.

Katie Atkinson and Trevor Bench-Capon. 2016. States, goals and values: Revisiting practical reasoning. *Argument & Computation* 7, 2-3 (2016), 135–154.

Pietro Baroni and Massimiliano Giacomin. 2009. *Argumentation in Artificial Intelligence*. Springer, Chapter Semantics of Abstract Argument Systems, 25–44.

Mairon Belchior, Jéssica Soares dos Santos, and Viviane Torres da Silva. 2018. Strategies for Resolving Normative Conflict That Depends on Execution Order of Runtime Events in Multi-Agent Systems. In *Proceedings of International Conference on Agents and Artificial Intelligence (ICAART 2018)*. SciTePress, 216–223.

Alexandros Belesiotis, Michael Rovatsos, and Iyad Rahwan. 2010. Agreeing on plans through iterated disputes. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*. IFAAMAS, 765–772.

Trevor Bench-Capon. 2003. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *Logic and Computation* 13, 3 (2003), 429–448.

Trevor Bench-Capon. 2016. Value-Based Reasoning and Norms. In *European Conference on Artificial Intelligence (ECAI 2016) (Frontiers in Artificial Intelligence and Applications)*, Vol. 285. IOS Press, 1664–1665.

Trevor Bench-Capon, Henry Prakken, and Giovanni Sartor. 2009. *Argumentation in Artificial Intelligence*. Springer, Chapter Argumentation in Legal Reasoning, 363–382.

Avrim L. Blum and Merrick L. Furst. 1997. Fast planning through planning graph analysis. *Artificial Intelligence* 90, 1 (1997), 281 – 300.

Rafael H. Bordini, Michael Wooldridge, and Jomi Fred Hübner. 2007. *Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology)*. John Wiley & Sons.

Gerhard Brewka and Thomas Eiter. 2000. Prioritizing Default Logic. In *Intellectics and Computational Logic*. 27–45.

Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zisheng Huang, and Leendert van der Torre. 2001. The BOID Architecture: Conflicts Between Beliefs, Obligations, Intentions and Desires. In *International Conference on Autonomous Agents (AGENTS 2001)*. ACM, 9–16.

J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. 2002. Goal generation in the BOID architecture. *Cognitive Science Quarterly* 2, 3-4 (2002), 428–447.

Maximiliano Celmo Budán, Mauro Javier Gómez Lucero, Carlos Iván Chesñevar, and Guillermo Ricardo Simari. 2015. Modeling time and valuation in structured argumentation frameworks. *Information Sciences*. 290 (2015), 22–44.

Martin Caminada. 2006. On the Issue of Reinstatement in Argumentation. In *Logics in Artificial Intelligence (LNCS)*, Vol. 4160. Springer, 111–123.

Martin Caminada and Leila Amgoud. 2007. On the evaluation of argumentation formalisms. *Artificial Intelligence* 171, 5-6 (2007), 286–310.

Martin Caminada, Roman Kutlak, Nir Oren, and Wamberto Weber Vasconcelos. 2014a. Scrutable plan enactment via argumentation and natural language generation. In *International conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2014)*. IFAAMAS/ACM, 1625–1626.

Martin Caminada, Sanjay Modgil, and Nir Oren. 2014b. Preferences and Unrestricted Rebut. In *Computational Models of Argument (Frontiers in Artificial Intelligence and Applications)*, Vol. 266. IOS Press, 209–220.

Laurence Cholvy and Frédéric Cuppens. 1995. Solving Normative Conflicts by Merging Roles. In *Proceedings of the 5th International Conference on Artificial Intelligence and Law (ICAIL '95)*. ACM, 201–209.

Frank S. de Boer, Koen V. Hindriks, Wiebe van der Hoek, and John-Jules Ch. Meyer. 2007. A verification framework for agent programming with declarative goals. *Journal of Applied Logic* 5, 2 (2007), 277–302.

Marina De Vos, Tina Balke, and Ken Satoh. 2013. Combining event-and state-based norms. In *International conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2013)*. IFAAMAS, 1157–1158.

Jéssica Soares dos Santos, Jean de Oliveira Zahn, Eduardo Augusto Silvestre, Viviane Torres da Silva, and Wamberto Weber Vasconcelos. 2018. Detection and Resolution of Normative Conflicts in Multi-agent Systems: A Literature Survey. In *Proceedings of International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018)*. ACM, 1306–1309.

Phan Minh Dung. 1995. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence* 77, 2 (1995), 321–358.

Xiuyi Fan and Francesca Toni. 2015. On Computing Explanations in Argumentation. In *AAAI Conference on Artificial Intelligence (AAAI 2015)*. AAAI Press, 1496–1502.

Sergio Pajares Ferrando and Eva Onaindia. 2017. Defeasible-argumentation-based multi-agent planning. *Information Sciences* 411 (2017), 1–22.

Richard E. Fikes and Nils J. Nilsson. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *International Joint Conference on Artificial Intelligence (IJCAI 1971)*. Morgan Kaufmann Publishers Inc., 608–620.

Dorian Gaertner and Francesca Toni. 2007. Preferences and Assumption-Based Argumentation for Conflict-Free Normative Agents. In *International Workshop on Argumentation in Multi-Agent Systems (LNCS)*, Vol. 4946. Springer, 94–113.

Alejandro Javier García, Carlos Iván Chesñevar, Nicolás D. Rotstein, and Guillermo Ricardo Simari. 2013. Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems. *Expert Systems with Applications* 40, 8 (2013), 3233–3247.

Michael Gelfond and Vladimir Lifschitz. 1988. The Stable Model Semantics for Logic Programming. In *ICLP/SLP*. MIT Press, 1070–1080.

Georgios K. Giannikis and Aspassia Daskalopulu. 2011. Normative conflicts in electronic contracts. *Electronic Commerce Research and Applications* 10, 2 (2011), 247–267.

Akin Günay and Pinar Yolum. 2013. Constraint satisfaction as a tool for modeling and checking feasibility of multiagent commitments. *Applied Intelligence* 39, 3 (2013), 489–509.

Koen V. Hindriks, Wiebe van der Hoek, and M. Birna van Riemsdijk. 2009. Agent programming with temporally extended goals. In *Autonomous Agents and Multiagent Systems (AAMAS 2009)*. IFAAMAS, 137–144.

Koen V. Hindriks and M. Birna van Riemsdijk. 2007. Satisfying Maintenance Goals. In *International Workshop on Declarative Agent Languages and Technologies (LNCS)*, Vol. 4897. Springer, 86–103.

Joris Hulstijn and Leendert W. N. van der Torre. 2004. Combining goal generation and planning in an argumentation framework. In *Non Monotonic Reasoning (NMR 2004)*. 212–218.

Thomas C. King, Marina De Vos, Virginia Dignum, Catholijn M. Jonker, Tingting Li, Julian Padget, and M. Birna van Riemsdijk. 2017. Automated multi-level governance compliance checking. *Autonomous Agents and Multi-Agent Systems* (2017), 1–61.

Martin Kollingbaum. 2005. *Norm-governed Practical Reasonig Agents*. Ph.D. Dissertation. University of Aberdeen.

Martin J. Kollingbaum and Timothy J. Norman. 2003. NoA - A Normative Agent Architecture. In *Joint Conference on Artificial Intelligence (IJCAI 2003)*. Morgan Kaufmann, 1465–1466.

Carmen Lacave and Francisco Javier Díez. 2004. A review of explanation methods for heuristic expert systems. *Knowledge Engineering Review* 19, 2 (2004), 133–146.

Hengfei Li, Nir Oren, and Timothy J. Norman. 2011. Probabilistic Argumentation Frameworks. In *Theory and Applications of Formal Argumentation (TAFA 2011)*. Springer, 1–16.

Tingting Li. 2014. *Normative Conflict Detection and Resolution in Cooperating Institutions*. Ph.D. Dissertation. University of Bath.

Felipe Meneguzzi, Odinaldo Rodrigues, Nir Oren, Wamberto Weber Vasconcelos, and Michael Luck. 2015. BDI reasoning with normative considerations. *Eng. Appl. of AI* 43 (2015), 127–146.

Sanjay Modgil. 2007. An Abstract Theory of Argumentation That Accommodates Defeasible Reasoning About Preferences. In *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (LNCS)*, Vol. 4724. Springer, 648–659.

Bernard Moulin, Hengameh Irandoust, Micheline Bélanger, and G. Desbordes. 2002. Explanation and Argumentation Capabilities: Towards the Creation of More Persuasive Agents. *Artificial Intelligence Reviwe* 17, 3 (2002), 169–222.

Nir Oren. 2013. Argument Schemes for Normative Practical Reasoning. In *Theory and Application of Formal Argumentation (LNCS)*, Vol. 8306. Springer, 63–78.

Nir Oren, Timothy J. Norman, and Alun D. Preece. 2007. Subjective logic and arguing with evidence. *Artificial Intelligence* 171, 10-15 (2007), 838–854.

Nir Oren, Sofia Panagiotidi, Javier Vázquez-Salceda, Sanjay Modgil, Michael Luck, and Simon Miles. 2008. Towards a Formalisation of Electronic Contracting Environments. In *Coordination, Organizations, Institutions and Norms in Agent Systems (COIN). (LNCS)*, Vol. 5428. Springer, 156–171.

Nir Oren, Wamberto Vasconcelos, Felipe Meneguzzi, and Michael Luck. 2011. Acting on Norm Constrained Plans. In *CLIMA (Lecture Notes in Computer Science)*, João Leite, Paolo Torroni, Thomas Ågotnes, Guido Boella, and Leon van der Torre (Eds.), Vol. 6814. Springer, 347–363.

Natalia Criado Pacheco. 2012. *Using Norms to Control Open Multi-agent Systems*. Ph.D. Dissertation. Universidad Politecnica de Valencia.

Sofia Panagiotidi, Javier Vázquez-Salceda, and Frank Dignum. 2012. Reasoning over Norm Compliance via Planning. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems (LNCS)*, Vol. 7756. Springer, 35–52.

J. Pitt, D. Busquets, and R. Riveret. 2013. Formal Models of Social Processes: The Pursuit of Computational Justice in Self-Organising Multi-Agent Systems. In *International Conference on Self-Adaptive and Self-Organizing Systems*. IEEE Computer Society, 269–270.

Henry Prakken. 2006. Combining sceptical epistemic reasoning with credulous practical reasoning. In *Computational Models of Argument (Frontiers in Artificial Intelligence and Applications)*, Vol. 144. IOS Press, 311–322.

Henry Prakken and Giovanni Sartor. 1997. Argument-Based Extended Logic Programming with Defeasible Priorities. *Journal of Applied Non-Classical Logics* 7, 1 (1997), 25–75.

Iyad Rahwan and Leila Amgoud. 2006. An Argumentation-Based Approach for Practical Reasoning. In *Argumentation in Multi-Agent Systems (LNCS)*, Vol. 4766. Springer, 74–90.

Anand S. Rao and Michael P. Georgeff. 1995. BDI Agents: From Theory to Practice. In *The first International Conference On Multi-Agent Systems*. 312–319.

Raymond Reiter. 1991. Artificial Intelligence and Mathematical Theory of Computation. Academic Press Professional, Inc., Chapter The Frame Problem in Situation the Calculus: A Simple Solution (Sometimes) and a Completeness Result for Goal Regression, 359–380.

Zohreh Shams, Marina De Vos, Nir Oren, and Julian Padget. 2016. Normative Practical Reasoning via Argumentation and Dialogue. In *International Joint Conference on Artificial Intelligence (IJCAI 2016)*. IJCAI/AAAI Press, 1244–1250.

Zohreh Shams, Marina De Vos, Julian Padget, and Wamberto Weber Vasconcelos. 2017. Practical reasoning with norms for autonomous software agents. *Engineering Application of AI* 65 (2017), 388–399.

Richmond H. Thomason. 2000. Desires and Defaults: A Framework for Planning with Inferred Goals. In *Principles of Knowledge Representation and Reasoning (KR 2000)*. Morgan Kaufmann, 702–713.

Alice Toniolo, Timothy J. Norman, and Katia P. Sycara. 2012. An Empirical Study of Argumentation Schemes for Deliberative Dialogue. In *European Conference on Artificial Intelligence (Frontiers in Artificial Intelligence and Applications)*, Vol. 242. IOS Press, 756–761.

M. Birna van Riemsdijk, Mehdi Dastani, and Michael Winikoff. 2008. Goals in agent systems: a unifying framework. In *Autonomous Agents and Multiagent Systems (AAMAS 2008)*. IFAAMAS, 713–720.

Wamberto Weber Vasconcelos, Martin J. Kollingbaum, and Timothy J. Norman. 2009. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems (AAMAS)* 19, 2 (2009), 124–152.

Javier Vázquez-Salceda, Huib Aldewereld, and Frank Dignum. 2005. Norms in multiagent systems: from theory to practice. *Computer Systems Science and Engineering* 20, 4 (2005).

Douglas N. Walton. 1996. *Argumentation Schemes for Presumptive Reasoning*. L. Erlbaum Associates.

Michael Wooldridge and Wiebe van der Hoek. 2005. On obligations and normative ability: Towards a logical analysis of the social contract. *Journal of Applied Logic* 3-4 (2005), 396–420.

Fabiola López y López, Michael Luck, and Mark d'Inverno. 2005. A Normative Framework for Agent-Based Systems. In *Normative Multi-Agent Systems (NMAS 2005)*. 24–35.